NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DESIGN AND DEVELOPMENT OF A
USER INTERFACE AND USER MANUAL
FOR A SYSTEM DYNAMICS MODEL
OF SOFTWARE MANAGEMENT

BY

DANIEL W. SWINDELL

MARCH 1989

Thesis Advisor: Tarek Abdel-Hamid

RITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

EPORT SECURITY CLASSIFICATION : Unclassified	1b RESTRICTIVE MARKINGS
ECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release Distribution is unlimited
ECLASSIFICATION/DOWNGRADING SCHEDULE	
RFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
AME OF PERFORMING ORGANIZATION aval Postgraduate School — 6b OFFICE SYMBOL (If applicable) 54	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
DDRESS (City, State, and ZIP Code) nterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
AME OF FUNDING SPONSORING RGANIZATION — 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

ITLE (Include Security Classification)
Design and Development of a User Interface and User Manual for a System Dynamics Model of Software Management

ERSONAL AUTHOR(S)
Swindell, Daniel W.

TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ____ TO ____	1989 March	153

UPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US. Govt.

COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	User interface, user friendly, batch language, User manual

BSTRACT (Continue on reverse if necessary and identify by block number)

Simulation models typically possess primitive user interfaces. Users must end substantial amounts of time learning the model before they gain oficiency in using the model. A versatile, user-friendly interface reduces e time and frustration involved in learning a new system. This research fort designs a user-friendly interface for the System Dynamics Model of ftware Management. The interface is written in a batch programming language, npatible with the IBM personal micro-computer. The interface is a product the prototype design approach. A sophisticated batch language provides the f-generating menu structures, advanced string handling capabilities, and or enhancements. The new user interface is now a valuable feature of the el and clearly shows the benefit of utilizing a prototype design approach this type of application.

ISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified	
NAME OF RESPONSIBLE INDIVIDUAL of. Tarek Abdel-Hamid	22b TELEPHONE (Include Area Code) (408) 646-2686	22c OFFICE SYMBOL Code 54AH

ORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

U.S. Government Printing Office 1986—606-243

Approved for public release; distribution is unlimited.

Design and Development of a User Interface and User Manual for a System Dynamics Model of Software Management

by

Daniel W. Swindell
Major, United States Marine Corps
M.E., Pennsylvania State University, 1988
B.S., Salem State College, 1980

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

ABSTRACT

Simulation models typically possess primitive user interfaces. Users must spend substantial amounts of time learning the model before they gain proficiency in using the model. A versatile, user-friendly interface reduces the time and frustration involved in learning a new system. This research effort designs a user-friendly interface for the System Dynamics Model of Software Project Management. The interface is written in a batch programming language, compatible with the IBM personal micro-computer. The interface is a product of the prototype design approach. A sophisticated batch language provides the self-generating menu structures, advanced string handling capabilities, and color enhancements. The new user interface is now a valuable feature of the model and clearly shows the benefit of utilizing a prototype design approach for this type of application.

C.1

TABLE OF CONTENTS

I. INTRODUCTION

A. PROBLEM STATEMENT

Computer proliferation in every facet of our society is occurring at an unprecedented rate. The cost of computer hardware continues to steadily decrease, making it possible for new applications to have more capable computers than ever before. The software to run these systems, however, has not enjoyed the same progress as the hardware. Large software projects have frequently failed to meet cost, schedule, and effectiveness goals. The failures of such large software development projects implies that the managers of the projects failed to understand the complex interrelationships of the hundreds of variables which effect a software project. The performance of the managers will naturally improve as they learn more about the process they must manage. Effective software estimation tools can be utilized to increase managerial effectiveness in two ways. First the tools provide managers with the requisite knowledge of the process and secondly they are an analysis tool which allow managers to assess the impact of changes to the process. Program simulation is an ambitious approach in software estimation to accurately depict the complex evolution of a software project. The use of simulations allow interactions to be represented which could not be otherwise accurately modelled.

The System Dynamics Model of Software Project Management is a software project simulation program. This simulation tool is uniquely based on systems dynamics concepts where feedback loops play an integral role. In the model, management and software forces interrelate in a dynamic fashion to reflect the software development process of the real world. As a training aid the model teaches managers the complex interrelationships of software project variables by providing quantitative results. This model, as powerful as it is, lacked the necessary user-friendly interface to attract a broad base of users.

B. PURPOSE OF RESEARCH

The purpose of this research is to design and develop an effective user-friendly interface for the model which would attract a broad base of users to the model. The interface design would substantially enhance the value of the model as a learning aid, as well as increase the user's productivity with the model.

C. SCOPE OF RESEARCH

The scope of the research includes the design and development of the user interface and the development of a user's manual to support the user interface. The research focuses on design considerations that facilitate rapid prototype development of a simple user environment. The research encompasses the current wisdom on user manual development. This research builds on earlier work done by Captain C. E. Haury, [Ref. 1].

D. THESIS ORGANIZATION

Chapter II addresses the design issues germane to the development of a user interface appropriate for this application. Chapter II covers popular guidelines for writing an effective user's manual. Chapter III explains the system architecture of the user interface. Chapter IV is the user's manual for the user interface.

II. USER INTERFACE DESIGN

A. FUNCTIONALITY

The user interface can only be effective in meeting user needs if the underlying system design provides the required functionality. If the system lacks essential capabilities, an interface will not be able to distract the user away from those deficiencies. If the system has the required capabilities but was poorly designed, again the interface can at best avoid further degradation of poor system performance. The underlying system functionality should, therefore, be understood as the foundation upon which the interface will be built.

B. HUMAN FACTORS GOALS

The particular user community and the set of tasks to be accomplished have a direct impact on the system design. The ability of the user community must be measured against the design goals and the identified tasks. There will typically be multiple design goals which can be traded off against each other. Examples of design goals are performance speed, error rate, learning time, retention time, and subjective user satisfaction. At the initial stage of design, the design goals will be formulated with the user community in mind. Once the tasks have been defined, a selective set of those tasks can be executed by a sample group of users from the user community to assess how well the design goals are being met. Such a group of tasks are often referred to as a benchmark set of tasks which are reflective of the characteristics of the larger group of tasks. If the design goals can not be met, then the tasks can possibly be altered or the design goals themselves altered. Knowledge of what exactly the design goals are and early testing of the tasks by users can greatly increase the likelihood of a successful design.

C. INTERACTIVE SYSTEM DESIGN GUIDELINES

The proliferation of personal computers coupled, with the evolution of mainframe terminals to include a display, have created an interactive computing environment. The

size, color and display rate of the displays obviously impacts the efficiency of the user. Shneiderman listed eight principles to adhere to during the design of interactive systems [Ref. 2:p. 61]. The eight principles are:

- Work for consistency
- Provide users with alternate short cuts
- Provide feedback to the user
- Design the closure property into actions
- Simplify error handling
- Allow easy reversal of actions
- Make the users sense they are in control
- Minimize short term memory requirements

Consistency is attained when similar situations require similar sequences of actions. The terminology and menu layouts should also be similar. An example of alternate short cut is the provision of a command language where the user has the option of entering the command mode and then entering a command. Feedback is some indication to the user reflecting the status of his request. The closure property here implies that sequences of actions are readily identifiable as having a beginning, middle, and an ending. The user should not be distracted by wondering where he is in a process. Error messages should clearly and concisely describe the error and the corrective action. Users having selected an option should have a convenient recourse to return to their previous position in the process. Unexpected outcomes are disruptive and damaging to the users confidence in the program and his ability to control the program. Menus should be kept uncluttered; the number of selections limited to a number which can be remembered easily.

D. ATTRACTING THE USER'S ATTENTION

There are many instances where the user's attention must be gained such as error conditions and situations which differ slightly from those he expects.

Shneiderman gave guidelines for some of the common means of attracting the user's attention:

- Intensity - up to two level
- Marking - underlining, boxes, bullets
- Size - up to four sizes
- Fonts - up to three fonts
- Inverse video
- Blinking - two to four hertz
- Color - up to four colors
- Audio - low tones for normal; loud tones for exceptions

E. ORDER OF MENU SELECTIONS

The order of the selections in a menu can have a significant effect on the time required to make a selection. An experiment by Card examined alphabetically, functionally, and randomly constructed menus. The experiment showed that if users knew what to scan for, their performance was best on alphabetically arranged lists and still high with functionally organized lists. The study group performed best on functionally arranged lists if they didn't know precisely what they were looking for. If the items to be on a menu follow a natural order such as chronological order or size, then that order should be used in the menu. If the items lack a natural order then the designer must decide what order to use. Frequency of use and relative importance are two commonly used orders. [Ref. 3:pp. 190-196]

F. DISPLAY RATE

The display rate is the rate information is displayed on the screen. In some situations such as telecommunications, or in intense calculations the display rate noticeably slows to the user. The slowness in those instances is due to the rate of the output to the display. A second factor affecting the display rate would be the actual capabilities of the display. A third factor would be the amount of information displayed per screen. The system designer can reduce the time to display a screen by making the screen contents as concise as possible.

G. PHRASING OF MENU ITEMS

Phrasing of menu items is not the same as the ordering of the items. Phrasing of the menu items addresses the semantics of the selection names. The wording of the selections can improve the readability and proper interpretation of the menu selections. Shneiderman suggests the following four criteria for effectively wording menu selections:

- Use familiar and consistent terminology
- Make items distinct
- Be concise and consistent
- Place the keyword to the left

The terminology should be familiar to the user and used consistently. Consistency between menus is important. By standardizing the placement of menu titles and instructions, colors, and text markings, the user can quickly adjust to each menu. [Ref. 2:pp. 110-115]

III. USER'S MANUAL

A. ON-LINE VERSUS HARD-COPY USER MANUALS

1. Introduction

Covington claimed that the most complained about aspect of a computer system is documentation [Ref. 4:p. 165]. He went on to assert that the software crisis of the 1970's has become the documentation crisis of the 1980's. Documentation is indisputably a vital element in a successful software product. Virtually all popular software packages are now delivered with paper documentation, which is referred to as hard-copy, and a variety of documentation in software form. The accompanying documentation will weigh heavily on both an interested customer's decision to purchase a product as well as his initial assessment of the product. The phrase on-line documentation is used here to include software that can be run and viewed separately from the main program, as well as actual on-line assistance.

2. On-line Documentation

a. Advantages

On-line documentation is generally more accessible than hard-copy. The documentation can be either placed in permanent memory or distributed widely on inexpensive disks. On-line documentation doesn't take up any appreciable office space. The user isn't daunted by the size of the documentation because of his limited view of the documentation. The cost and time to update the documentation is small compared to hard-copy.

3. Hard-copy Documentation

a. Advantages

The user doesn't have to leave his location in the environment to access the documentation. Hard-copy is easier to track and control as opposed to on-line which can proliferate easily. Hard-copy is less fatiguing to read and yields greater comprehension rates. Gould and Grischkowsky conducted an experiment where thirty-two subjects read material in equivalent twenty-three line pages of hard-copy and

on-line media [Ref. 5]. The reading rate was significantly higher for the hard-copy as well as the comprehension accuracy. Hard-copy has enjoyed both color and resolution advantages over the on-line medium. Display graphics are making enormous strides as processing power, memory limits and display technology continue to be pushed ahead; but the quality and variety of display options still resides with hard-copy documentation.

B. DOCUMENTATION DEVELOPMENT

1. Fundamental Principles

As in any writing, simple adherence to basic writing style guidelines will make the difference in the readability of the material. Sentences, paragraphs, and chapters should be laid out thoughtfully. Today the writer can take advantage of spelling checkers to correct his spelling errors, grammar, checkers to check his grammar and style evaluators to evaluate his writing style throughout his document. These tools can not be relied on to catch and correct all errors but they are a major benefit to most writers. Proof reading is an indispensable phase of any writing. As with software, documentation should be tested on a sample of intended users.

2. Software Development Life Cycle

The documentation will obviously have to be complete and delivered with the software package. The impact of poor documentation was touched upon earlier. The documentation will continue to require resources to correct errors and incorporate updates resulting from modifications to the software. The not so obvious questions are what type of resources will be required to meet the delivery deadline and when should those resources be dedicated? Adding to the intricacy of this question is the trade off between the level of product support required after production and the quality of the documentation. Anne Hedin poignantly asks, "Whether it's another department on the phone to you, or your people on the hotline to a vendor, can you afford to substitute downstream support for up-front, user-friendly documentation?" [Ref. 6:p. 37] The documentation effort needs to begin early in the development process. The documentation effort can in fact help to clarify ambiguities in the specification which might otherwise go unnoticed until later in the development process. The

documentation should, like the software it is based on, fall under a careful schedule that includes timely reviews.

3. Documentation Forms

There are a number of prevalent forms of documentation in the popular software packages that fall under various names: owner's manual, operating manual, reference manual, tutorials, command summary cards. Basically they can be divided into three categories: the operating manual, the reference manual and the command summary card. The operating manual is the document that leads the user into the system and should be designed to present the user with a view of the system at a high level. At a high level, the details are hidden from him because he doesn't need them at that time. The reference manual has all the details which the experienced user may at some time require. The command summary card is a short summary of the command formats to refresh the mind of the user. Tutorials are another form of documentation which often stand on their own. Tutorials painstakingly lead the user through the sequences of the system.

4. Documentation Content

a. Organization

(1) Task Organization. One trend in user documentation is task organization. Task organization is the structuring of the manual around the tasks the user will perform. The documentation must be written from the viewpoint of the user as he will face the tasks. [Ref. 6]

(2) Top Down Design. Top down design is characterized by successive levels which begin with a high level view and few details and progress to the lower levels with greater detail. Covington asserts that good programmers can be good writers because both programming and writing documentation require top down design. The writer must keep track of the readers knowledge to successfully implement top down design. This means he must remember what knowledge the reader began with and what knowledge he has revealed up to that point in the document. [Ref. 4]

b. Style

(1) Script Format. When the document is task organized the user is presented with tasks. The tasks can be addressed using a script which the user follows. The user is told that if he wants a result he should perform the following actions and he will see a certain result.

(2) Examples. Examples convey quickly what would take much longer to explain using prose and therefore should be used freely. Concepts, however, usually can not be conveyed by a an example that the user will see on the screen and require prose. Shneiderman is a strong advocate of utilizing the conceptual knowledge the user already possesses as the foundation for quickly comprehending related computer concepts [Ref. 2:p. 49]. He writes that an effective strategy is to begin with a known concept, translate that concept into the general computer arena, and finally over to how the concept applies in a particular system.

(3) Nonanthropomorphic style. Beginning users initially consider references to the system as possessing human qualities as novel. However, as users gain experience they become irritated by those references. The writer should instead focus on the user and the tasks he needs to accomplish. [Ref. 2:p. 370]

IV. SYSTEM ARCHITECTURE

A. SYSTEM OVERVIEW

This chapter discusses the system architecture of the System Dynamics Model of Software Project Management. The system consists of four modules: the user interface, the Dynex interface, the Professional Dynamo environment, and the Dynamica model. The high level view of the system architecture is depicted in Figure 1.

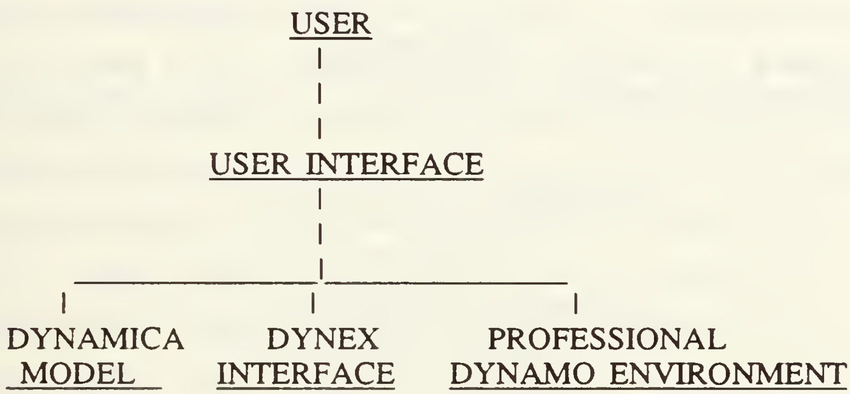


Figure 1. System Overview

The user encounters the user interface upon starting the system. The user interface calls upon the other three system components to execute the user directed tasks. The user is returned to the user interface after each task is completed. The user interface carries out user selections by issuing the appropriate Professional Dynamo command along with the associated Dynamica model file name as an argument. Professional Dynamo is the underlying simulation language in which the Dynamica Model is written [Ref. 7]. The user can elect to leave the user interface entirely and work directly with the Professional Dynamo environment. The user will be returned to the user interface when he exits the Professional Dynamo environment.

If the user desires to change the values of the variables in the current model the user interface transfers the user to the Dynex model interface to make those changes. He will be returned to the user interface after making his desired changes.

B. USER INTERFACE

The user interface is a menu driven shell that presents the user with a simple and logical view of the system. A shell is a hospitable user working environment provided by a program to enhance the basic operating system environment. This particular shell is generated by a sophisticated batch file which, once executed by the user at start-up, controls the execution of the other executable files. The shell program was written using Extended Batch Language (EBL). EBL is used to write batch files that possess enhanced string handling capabilities, advanced control structures, and superior screen control over those written in ordinary batch file constructs [Ref. 8:p. 2]. Batch files, enhanced by EBL, were selected as the basis of the shell because a batch file was simpler to write than using a high level language. This approach allowed a rapid development of a prototype interface which could be easily modified. EBL provided color and border menu enhancements without any significant development cost. The shell includes automatic return to the calling menu, optional immediate return to the calling menu and user input range checking.

The main menu, as seen in Figure 2, offers nine options which distinguish the primary system functions of system help, model variable changes, model simulations, viewing results, storing results, and transferring to the Professional Dynamo environment.

Five of the nine main menu options produce sub-menus with multiple options, as depicted in Figure 3. The four options which are not sub-menus drive tasks which are carried out directly by the main menu. The View Variable Plots option and the Professional Dynamo option menus have different formats than the other sub-menus because the user interface environment must call Professional Dynamo to accomplish those tasks. Three of the sub-menus themselves possess subordinate sub-menus. This third level of menu control is also depicted in Figure 3.

MAIN MENU

THE DYNAMICA MODEL OF SOFTWARE PROJECT MANAGEMENT

1 MODEL REQUIREMENTS

2 HELP FACILITIES

3 SET MODEL VARIABLES

4 RUN SIMULATIONS

5 VIEW VARIABLE PLOTS

6 VIEW STANDARD PLOTS

7 STORE RESULTS

8 PROFESSIONAL DYNAMO INTERFACE

9 EXIT TO DOS

Choose an option: (ESC exits menu):

Figure 2. Main Menu

```
                              MAIN MENU
                                  |
       -------------------------------------------------------
       |            |             |             |            |
   OPTION 2     OPTION 3      OPTION 5      OPTION 6     OPTION 8
       |            |             |             |            |
       |            |             |             |            |
    HELP         SET           PROF          VIEW          PROF
    MENU         MODEL         DYNAMO        STANDARD      DYNAMO
       |         VARIABLES     VIEWER        PLOTS         MAIN
       |         MENU          MENU          MENU          MENU
       |            |_ _ _ _      _ _
       |            |               |
   OPTION 3     OPTION 2      OPTION 3
       |            |             |
       |            |             |
    PROF         DEFINITIONS    SET
    DYNAMO       OF             MODEL
    HELP         VARIABLES      VARIABLES
    MENU         MENU           MENU
                                (in PD)
```

Figure 3. Menu Structure

The shell in a larger sense contains five batch files: the main program and four subservient batch files, Figure 4. The four batch files control the system help, the changing of model variable values, the plotting of simulation results, and the storing of results.

MAIN BATCH FILE	SUBSERVIENT BATCH FILES
Menu.Bat	Help.bat
	Variabl.bat
	Plot.bat
	Storstat.bat

Figure 4. Shell Bat-file Breakdown

The main batch program, menu.bat, is always returned to after the other batch files terminate. Menu.bat runs the bat.com file and displays the main menu. Bat.com loads the instructions into memory which define the EBL commands in a batch file. Batch files can then distinguish EBL commands from normal DOS system commands by the key word "bat" which precedes the EBL commands. The EBL EXECMAKE function was used to generate the control structure of menu.bat. The file hierarchy structure is presented in Figure 5. Figure 5 shows menu.bat calling bat.com, help.bat, variabl.bat, plot.bat, and storstat.bat. The file, help.bat, provides an introduction to the model, a summary of the on-line help, and instructional explanations of the features of Professional Dynamo. Eight Professional Dynamo help topics are stored separately as text files. The help.bat file executes the list.com program, which uses the desired text help file as an argument to list the desired help. The eight help files have the characteristic ".hp" ending.

The file, variabl.bat, is called by the main batch file to manage the alteration of the adjustable model variables. Variabl.bat displays the names, abbreviations, and definitions of those adjustable variables. Variabl.bat calls the file dynex.exe to actually change the variables.

```
          ***BAT.COM
          ***HELP.BAT
          *     *
          *     ***LIST.COM
          *          *          *PD.HP
          *          *          *DYNEX.HP
          *          *          *EDTR.HP
          *          **********SMLT.HP
          *                     *TOOLS.HP
          *                     *TRNS.HP
          *                     *UTILS.HP
          *                     *VIEW.HP
          ***VARIABL.BAT
          *      *
          *      ***DYNEX PROJECT.DNX
MENU.BAT**
          ***SMLT PROJECT *GO=
          ***VIEW PROJECT.RSL
          ***PLOT.BAT
          *      *              *REP PROJECT PLOT1.DRS
          *      **************REP PROJECT PLOT2.DRS
          *                     *REP PROJECT PLOT3.DRS
          *                     *REP PROJECT PLOT4.DRS
          *
          ***STORSTAT.BAT
          *      *
          *      ***REP PROJECT STATS
          *
          ***PD PROJECT.DYN
                 *
                 ***INSERT
```

Figure 5. File Hierarchy

The file, plot.bat, is called to manage the viewing of standard plots. Plot.bat offers an explanation of the standard plot options and a selection of four predefined plots.

The file, storstat.bat, is called to managed the storing of simulation results. Storstat.bat queries the user for a name and path to store the results under before saving the simulation results.

SMLT.exe, view.exe and PD.com are Professional Dynamo files which are also executed directly by menu.bat. SMLT.exe runs the simulation. View.exe allows the user to define his views of the simulation variables. PD.com allows the user to directly access the Professional Dynamo environment. The "project" arguments are the Dynamica model files used by the Professional Dynamo files.

C. DYNAMICA MODEL

The Dynamica model is a set of files created in the Professional Dynamo environment. The primary model file has the ".dyn" ending. The user should not alter the model itself. The user interface, in addition to guiding the user through his tasks, shelters the model from inadvertent changes by the user.

D. DYNEX INTERFACE

The Dynex.exe program is provided by Professional Dynamo to allow the construction of a user friendly interface. Using Dynex.exe the user can interactively change the value of model variables. The seventeen variables, see Figure 6, which can be changed are predefined in the file project.dnx; project.dnx is the argument file used by dynex.exe. The names and definitions of the seventeen variables are contained in the variable.bat file.

ADJUSTABLE VARIABLES

1. RJBDSI . . . . Real Job Size in DSI
2. DSIPTK . . . Delivered Source Instruction Per Task
3. TNERPK. Error Rate Per 1000 Delivered Source Instr
4. HIREDY . . . . Hiring Delay
5. ASIMDY . . . . Assimilation Delay
6. AVEMPT . . . . Average Employment
7. UNDEST . . . . Task Underestimation Factor
8. TOTMD1 . . . . Total Mandays
9. TDEV1 . . . . Time to Develope
10. DEVPRT % of Effort Assumed Needed For Development
11. TPFMQA . .Fraction of Manpower Devoted to Quality Assurance
12. INDUST . . . . . . Initial Understaffing Factor
13. WCWF1 . . Willingness to Change the Workforce
14. TRPHNR . .% Experienced Employee Effort to Train a New Employee
15. AMPPS ....Average Daily Manpower / Staff Expended On Project
16. MAXLEN . . . . Max Length Of Project
17. SAVPER . . . . Save Period For Data

Figure 6. Adjustable Variables

As seen in Figure 5, menu.bat calls variable.bat which in turn calls dynex.exe with project.dnx as its argument. Dynex.exe also uses project.ins, project.smt, project.dat, and project.was. The variable values are stored in project.dat. The current values are displayed to the user and then stored again after the user makes his changes. Whereas the current values only are stored in the project.dat file, project.was is used to capture the old and new values of the change as a permanent record for the user.

E. PROFESSIONAL DYNAMO ENVIRONMENT

The user can elect to depart the user interface and work directly in the Professional Dynamo Environment by selecting this option in the main menu. Menu.bat executes PD.com, with project.dyn as its argument, to place the user in the Professional Dynamo Environment. The file "insert" is required for PD.com to execute. Help files which provide explanations of the Professional Dynamo features are provided in the user interface help section. As stated earlier, the user could inadvertently alter the model itself once inside the Professional Dynamo Environment and should, therefore, consult the Professional Dynamo user manual before exercising this option.

F. RUNNING SIMULATIONS

The user interface runs the simulation by executing the Professional Dynamo file, SMLT.exe. SMLT.exe requires the three compiled files project.ins, project.dat, and project.smt to operate on. The SMLT.exe file accepts the name of the output file as its argument. The".rsl" extension is automatically attached to the argument name.

G. VIEWING RESULTS

The main menu offers two selections, View Variable Plots and View Standard Plots, for viewing simulation results. The View Variable Plots option provides the user the flexibility of selecting the variables he would like displayed. This option also provides the capability of displaying the results in tabular or graphical formats.

The user interface displays the results of a simulation by executing the Professional Dynamo file, view.exe. View.exe accepts as an argument a file with the model name and the extension ".rsl". The user is presented the Professional Dynamo screen for viewing simulation results. The user can then choose either the tabular or graphical displays. If he chooses graphical displays, he is limited to six variables per display.

The View Standard Plots selection only provides a pre-defined set of points for display without the flexibility of choosing the variables to be displayed. The plot.bat file presents the user with a choice of four predefined graphs for display. Plot.bat executes the Professional Dynamo file, rep.exe, which displays the desired graph. Rep.exe requires two files as arguments: a file with the ".rsl" extension and a file with the ".drs" extension. The ".drs" file is the report specification file; in the plot.bat file the ".drs" files are plot1.drs, plot2.drs, plot3.drs and plot4.drs.

H. STORING RESULTS

The user interface stores simulation results by calling the Storstat.bat batch file. Storstat.bat initially executes rep.exe to display the simulation results according to the format specified in the "stats" file. Storstat.bat then queries the user for the name of the file in which to store the simulation results and for the pathname of the directory in which to store the output file. Existing files under that name, in the specified directory, are overwritten.

V. DYNAMICA USER'S MANUAL

A. INTRODUCTION

The Dynamica Model is a software development process model which is based on the concepts of system dynamics. The model integrates both the management-type functions, such as planning, controlling and staffing, along with the traditional software production-type functions, such as design, coding, quality assurance, and testing. The application of system dynamics concepts enables the model to represent the fundamental feedback processes, where functions interact and affect other functions in the development process. The model portrays the effect of the feedback relationship through simulation execution. [Ref. 9:pp. 6-11]

B. DYNAMICA MODEL REQUIREMENTS

1. Memory

The model requires an IBM or IBM compatible XT/AT microcomputer with 640 K of RAM with a hard disk drive or a 1.2 Meg floppy disk drive.

2. Co-processor

A math co-processor, although not necessary, will dramatically reduce the simulation run-time.

3. Displays

The model supports monochrome, MDA, CGA and EGA graphic display modes.

4. Software Requirements

The model requires MS DOS or PC DOS version 2.xx or later.

C. BACKGROUND

The Dynamica model was developed to accurately represent the complicated software development process. A secondary objective was to create a learning tool with which managers could gain understanding of the process they must manage. The combination of integrative design and simulation capability allow users of the

model to experiment with different variables in the development process and quickly see the results of their experimentation.

The Dynamica model was written in Professional Dynamo, a simulation design language. The model is based on four sub-systems: Human Resource Management, Software Production, Controlling and Planning. A high level view of the Dynamica model which illustrates the interaction between the four sub-systems is shown in Figure 7 [Ref. 10:p. 12].

The human resource management sub-system depicts hiring, training, assimilation, and transfer of human resources. The software production sub-system depicts the design, coding, quality assurance, rework, and testing activities. The planning sub-system reflects the scheduling activities of the project's life cycle. The control sub-system records the progress of the project. [Ref. 10:pp. 11-25]

Professional Dynamo has a rudimentary facility, called Dynex, for enhancing models by providing the capability of interactively adjusting the values of the model variables. Dynex was used to make seventeen of the Dynamica model variables accessible to the user. The user can now change the value of those variables to suit the software project being modelled. The system shell insulates the user from the dynamica model by providing an easy to use menu structure and enhanced screen displays. The shell was written using a software product called Extended Batch Language.

D. STARTING THE DYNAMICA MODEL

The model is started by issuing the command "Menu" from the directory where the Dynamica files are stored. The main menu will then appear, see Figure 8. The main menu options are numbered so that selecting the desired number will activate that portion of the model. The user may exit the model by either hitting the "esc" key or by selecting option 9. Option 9 will clear the screen before returning the user to the operating system whereas hitting the "esc" key will not clear the screen before returning the user to the operating system.

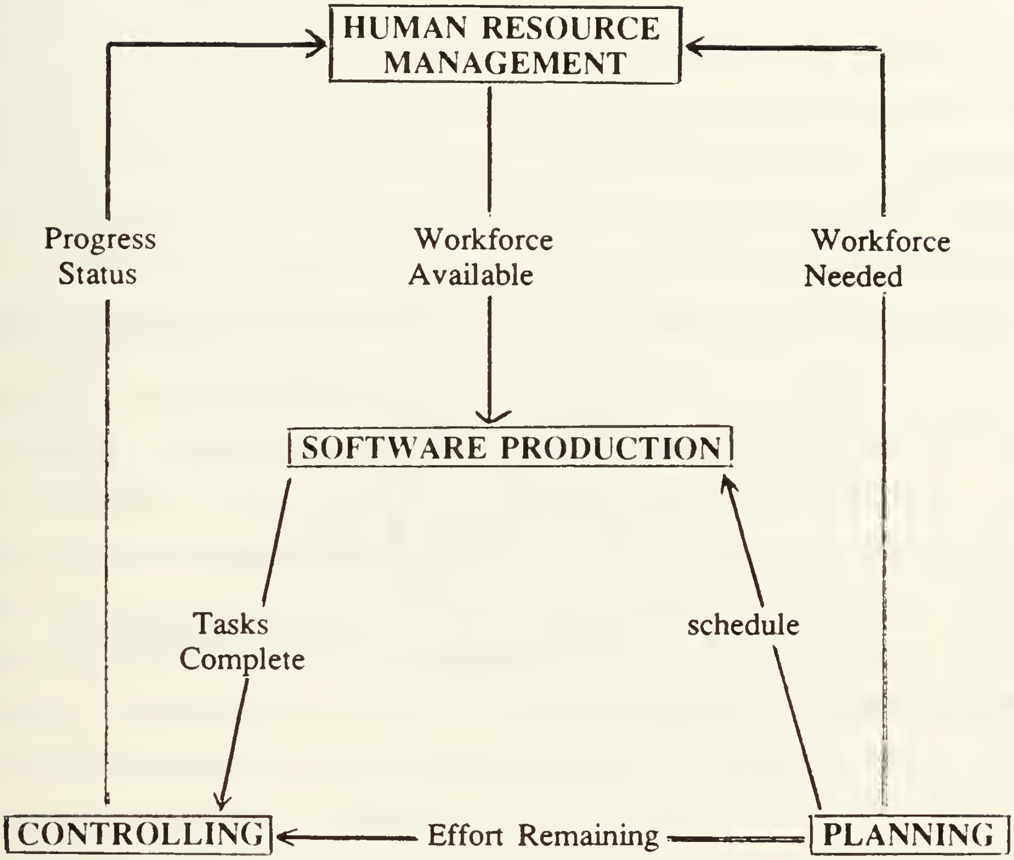


Figure 7. Dynamica High Level View.

MAIN MENU
THE DYNAMICA MODEL OF SOFTWARE PROJECT MANAGEMENT

1 MODEL REQUIREMENTS
2 HELP FACILITIES
3 SET MODEL VARIABLES
4 RUN SIMULATIONS
5 VIEW VARIABLE PLOTS
6 VIEW STANDARD PLOTS
7 STORE RESULTS
8 PROFESSIONAL DYNAMO INTERFACE
9 EXIT TO DOS

Choose an option: (ESC exits menu):

Figure 8. Main Menu

E. MAIN MENU

The menu system presents a simple and logical view of the Dynamica model to the user. The system is designed to start at the main menu and return the user to the main menu after each selection is completed. The main menu, as seen in Figure 8, offers nine options which distinguish the primary system functions of system help, model variable changes, model simulations, viewing results, storing results, and transferring to the Professional Dynamo environment.

Five of the nine main menu options produce sub-menus with multiple options, as depicted in Figure 9. The four options which are not sub-menus drive tasks which are carried out directly by the main menu. The View Variable Plots option and the Professional Dynamo option menus present slightly different formats than the other sub-menus because the user interface environment must call Professional Dynamo to accomplish those tasks.

F. HELP INFORMATION

Selecting the "help" option of the main menu causes the help menu in Figure 10 to be presented. Option (1) of the help menu, Model Introduction, is a one page explanation of the Dynamica model. Option (2) of the help menu, On-line Help, is a single screen explaining the potential for on-line help and the limited availability. Presently there is only one screen installed and it is in the "Set Variables" section of the program. Selecting option (3) of the help menu, Professional Dynamo Help, results in an intermediate screen appearing which explains that the following help is intended for experienced users who are familiar with Professional Dynamo. The Professional Dynamo help is provided to support the user who elects to work directly in the Professional Dynamo environment by selecting the Professional Dynamo option of the main menu.

```
                              MAIN MENU
                                  |
     -------------------------------------------------------------
     |            |               |               |              |
 OPTION 2     OPTION 3        OPTION 5        OPTION 6       OPTION 8
     |            |               |               |              |
     |            |               |               |              |
  HELP         SET             PROF            VIEW           PROF
  MENU         MODEL           DYNAMO          STANDARD       DYNAMO
     |         VARIABLES       VIEWER          PLOTS          MAIN
     |         MENU            MENU            MENU           MENU
     |            |_ _ _ _      _ _
     |            |               |
 OPTION 3     OPTION 2        OPTION 3
     |            |               |
     |            |               |
  PROF        DEFINITIONS      SET
  DYNAMO      OF               MODEL
  HELP        VARIABLES        VARIABLES
  MENU        MENU             MENU
                               (in PD)
```

Figure 9. Menu Structure

The Professional Dynamo help menu, see Figure 11, offers an overview of the available help and seven options for help on Professional Dynamo features.

HELP FACILITIES

1. MODEL FACILITIES
2. ON-LINE HELP
3. PROFESSIONAL DYNAMO HELP

Choose an option: (ESC exits menu):

Figure 10. Main Help Menu

Each of the help sections on Professional Dynamo is actually an individual text file. An editor lists the desired file for viewing. The menu bar at the bottom of the screen describes the commands active for viewing the help file. The user exits that help file and returns to the Professional Dynamo help menu by hitting the "esc" key, the letter "Q" or the letter "X". Selecting "?" from within the help editor displays the available features of the editor for viewing the help files. The user is returned to the Professional Dynamo help menu upon exiting a help file viewing session.

G. CHANGING THE VALUE OF MODEL VARIABLES

The Dynamica model designer has provided the user with access to seventeen of the model variables, see Figure 12. The user can review the definitions of the seventeen variables and then change the value of any of those variables by selecting the "Set Model Variables" option of the main menu. Selecting this option causes the Set Model Variables menu, Figure 13, to be displayed.

Selecting option (1), List Variables, of this menu causes a two page list of the seventeen variables, Figure 14, to be displayed. The variables were assigned to the following four categories for ease of access: Actual Job Size, Variables Characteristic of the Organizational Environment, Policy Variables, and Model Control. These categories are used to view the variable definitions and to change the variable values.

PROFESSIONAL DYNAMO HELP

THE FOLLOWING MENU PROVIDES ACCESS TO PD HELP FACILITIES

1. OVERVIEW OF PD HELP
2. DYNEX HELP
3. EDITOR HELP
4. SIMULATION HELP
5. TOOLS HELP
6. TRANSLATOR HELP
7. UTILITIES HELP
8. VIEW HELP

Choose an option: (ESC exits menu):

Figure 11. Professional Dynamo Help Menu

ADJUSTABLE VARIABLES

1. RJBDSI . . . . Real Job Size in DSI
2. DSIPTK . . . Delivered Source Instruction Per Task
3. TNERPK. Error Rate Per 1000 Delivered Source Instr
4. HIREDY . . . . Hiring Delay
5. ASIMDY . . . . Assimilation Delay
6. AVEMPT . . . . Average Employment
7. UNDEST . . . . Task Underestimation Factor
8. TOTMD1 . . . . Total Mandays
9. TDEV1 . . . . Time to Develope
10. DEVPRT % of Effort Assumed Needed For Development
11. TPFMQA . Fraction of Manpower Devoted to Quality Assurance
12. INDUST . . . . . . Initial Understaffing Factor
13. WCWF1 . . Willingness to Change the Workforce
14. TRPHNR .% Experienced Employee Effort to Train a New Employee
15. AMPPS . Average Daily Manpower / Staff Expended On Project
16. MAXLEN . . . . Max Length Of Project
17. SAVPER . . . . Save Period For Data

Figure 12. Adjustable Variables

SET MODEL VARIABLES

1 LIST VARIABLES

2 DEFINITIONS OF VARIABLES

3 SET VARIABLES

Choose an option: (ESC exits menu):

Figure 13. Model Variable Menu

LISTING OF VARIABLES

The following 17 model variables of Dynamica are adjustable via the menu.
They are grouped in 4 categories for easy access.

I. ACTUAL PROJECT SIZE
   1. RJBDSI . . . . Real Job Size in DSI

II. VARIABLES CHARACTERISTIC OF THE ORGANIZATIONAL ENVIRONMENT
   A. Productivity
      2. DSIPTK . . . Delivered Source Instruction Per Task
   B. Quality
      3. TNERPK. Error Rate Per 1000 Delivered Source Instr
   C. Staffing Variables
      4. HIREDY . . . . Hiring Delay
      5. ASIMDY . . . . Assimilation Delay
      6. AVEMPT . . . . Average Employment

III. POLICY VARIABLES
   A. Estimation
      7. UNDEST . . . . Task Underestimation Factor
      8. TOTMD1 . . . . Total Mandays
      9. TDEV1 . . . . Time to Develope
   B. Resource Allocation
      10. DEVPRT % of Effort Assumed Needed For Development
      11. TPFMQA . Fraction of Manpower Devoted to Quality Assurance
      12. INDUST . . . . . . Initial Understaffing Factor
      13. WCWF1 . . Willingness to Change the Workforce
   C. Staffing
      14. TRPHNR . . . % of Experienced Employee Effort to Train a New Employee
      15. AMPPS . Average Daily Manpower / Staff Expended On Project

IV. MODEL CONTROL
   16. MAXLEN . . . . Max Length Of Project
   17. SAVPER . . . . Save Period For Data

Figure 14. Adjustable Variable Selection Menu

Each variable, except for the Real Job Size in DSI, is listed under a label which indicates what area of the model the variable falls. The variable abbreviation and full name are then given.

Selecting option (2), Model Definitions, causes the Model Definitions menu to appear, see Figure 15. Selecting one of the four variable categories results in the displaying of the definitions of the variables in that category. After selecting a category for viewing, the user must page through the definitions of that category in order to return to the Model Definitions menu. The user may return to the Set Model Variables menu from the Model Definitions menu by hitting the "esc" key.

Selecting option (3), Set Variables, of the Set Model Variables menu, causes the Set Model Variables/ Dynex Menu to be displayed, see Figure 16. This menu is again organized by the same four previous categories. The user can select option (5) to return to the previous menu if he accidentally chose the Set Variables menu. The user can select whichever categories he would like to edit by typing the category numbers separated by commas or spaces and then pressing "enter". Selecting a category allows the user to change the values of any of the variables in that category. Each variable in that category is displayed with its currently assigned value. The user may enter a new value and hit "enter" to assign a new value to the variable. The user may alternately accept the presently assigned value by simply pressing "enter". The next variable in that category is then automatically displayed for editing. The variables of the next category selected for editing are automatically presented when the previous category has been edited. When the last category has been edited the user is returned to the Set Model Variables menu. The user can at that point again choose the Set Variables option to make further changes or corrections to the model variables. Changes made to the model variables are saved so that the next time the model is used the new values are displayed.

H. RUNNING A SIMULATION

Selecting option (4), Run Simulations, of the main menu causes the Dynamica model simulation to run. A running tally of the run-time is displayed in the lower right corner of the screen.

DEFINITIONS OF VARIABLES

1 ACTUAL PROJECT SIZE

2 ORGANIZATIONAL ENVIRONMENT VARIABLES

3 POLICY VARIABLES

4 MODEL CONTROL VARIABLES

Choose an option: (ESC exits menu):

Figure 15. Definitions of Variables Menu

SET MODEL VARIABLES
DYNEX MENU

1. ACTUAL PROJECT SIZE
2. ORGANIZATIONAL ENVIRONMENT VARIABLES
3. POLICY VARIABLES
4. MODEL CONTROL VARIABLES
5. RETURN TO MAIN MENU

Enter the number(s) of your selected choices.
(Separate each choice by a space or a comma.)

Figure 16. Dynex Set Model Variables Menu

The simulation uses the most recent variable values which are saved in a file. The results of the simulation are stored in a file and the user is returned to the main menu. The user may then view the simulation results via options (5) or (6) of the main menu.

I. VIEWING THE RESULTS OF A SIMULATION

1. Options

The results of a simulation may be viewed via option (5), View Variable Plots, or option (6), View Standard Plots, of the main menu.

2. View Variable Plots Option

The View Variable Plots option provides the flexibility of specifying up to six of thirty variables for display on the same graph; see Figure 17 on the following page. The user may alternately choose to see a tabular presentation instead of a graphical presentation. After the user selects option (5) of the main menu the user is asked if the graph should be displayed in Enhanced Graphics Adaptor(EGA) mode or in the monochrome mode. Following either response to the display mode question, Figure 18 appears.

"Viewing Project: Plot Select_Tabulate Tabulate_All Old Help Esc Quit"

Figure 18. Viewing Project Menu

"Plot" is initially highlighted but the highlight can be moved to any of the other options by using the keyboard arrows. The highlighted option can be executed by hitting "enter". An alternate method of executing an option is to type the capital letter of the option. Selecting either option "Esc" or "Quit" returns the user to the main menu. A view session is active until the user returns to the main menu. Selecting option "Old" results in the user being prompted for the number of the view he would like to display. The view must have been created during the present view session. If the user has just begun the view session and not created a view yet a "beep" will sound. Upon leaving the Viewing menu, by escaping or quitting to the main menu, the views defined during that session are lost as well as your choice of display mode.

LIST OF OUTPUT VARIABLES

TMPRMR	DEVMD	TDEV	TOTMD	TSTMD	ASSPRD	COMMOH
FTEQWF	MPPTPD	PDEVRC	PMDSHR	POTPRD	PRCTDT	PRDPRD
TOTWF	AFMDPJ	CMERES	CMQAMD	CMRWET	CMRWMD	CMTKDV
CUMERG	CUMMD	CUMTKT	JBSZMD	PBJSZ	SCHCDT	
FRWFEX	SDVPRD	CMTSMD				

Figure 17. List of Output Variables

Selecting the "Plot" option of the Viewing menu results in the abbreviations of thirty variables being displayed. A blinking cursor can be moved to any of the variables by using the keyboard arrows. A variable is selected for display by moving the cursor to it and highlighting it by hitting "enter". Six of the thirty variables may be selected for any one display. The additional variables, beyond the seventeen the user may adjust, are other model variables that may be plotted but not adjusted. The selection process is terminated by hitting "enter". If the EGA mode is chosen, the selected variables are plotted in different colors and a color coded legend of the variables is displayed at the top of the graph. Once the variables are plotted, the menu in Figure 19 appears at the bottom of the graph display:

"View #: Next view_no Print Esc Quit"

Figure 19. View # Menu

Each time the user selects "Plot", "Select_Tabulate", or "Tabulate_All" the next consecutive view number is assigned to that view. The "#" will reflect the assigned view number. Selecting "Next" causes the next consecutively numbered view to be displayed. "View_no" is actually part of "Next" and not an option itself. Choosing "Esc" by hitting the "esc" key returns the user to the Viewing menu. Selecting "Quit" returns the user to the main menu.

Selecting the "Select_Tabulate" option of the Viewing menu allows the user to display selected variables in a tabular format. The thirty variable abbreviations are displayed for selection. The selection process is the same as described in the "Plot" process however, the user is not limited to six variables. The selection process is completed by hitting the "esc" key. The user is then prompted to enter the display interval. The selected variables are then displayed in accordance with the given interval. The "Next", "Esc", and "Quit" options are described above in the Viewing section.

The "Tabulate_All" option of the Viewing menu displays the values of all thirty of the listed model variables. The user is again prompted for the display interval before the variables are displayed. The "Next", "Esc", and "Quit" options are described above in the Viewing section.

3. View Standard Plots Option

Selecting option(6) of the main menu results in the View Standard Plots Menu appearing, Figure 20. The user can select one of the four pre-defined plotting functions by simply selecting the menu option number. After selecting the desired plot, the plot number and its variables will be displayed so that the user can see the full names of the variables. Figure 21 shows the variables by plot number; the appropriate set will be displayed before the plot is made. The user is also asked on this screen to type "Y" if he has an EGA monitor or "N" if he has a monochrome monitor. The variables associated with the particular plot will then be plotted together on the screen. The display will be in either monochrome or EGA, depending on the user's declaration. Selecting either "Esc" or "Q" from this plot screen will return the user to the "View Standard Plots" menu. The "Next view" option is inoperative on the standard plot screens.

J. SAVING SIMULATION RESULTS

The results of a simulation are saved by selecting option (7), Storing Results, of the main menu. The results are first displayed for inspection, as shown in the example in Figure 22. As seen in Figure 23, the information to be saved is overall project statistics. The user is prompted to provide a filename and a directory path to save the results under. Any previous file under the given filename will be replaced by the new results file. If the user hits "enter" without a pathname, the file is stored in the default directory.

PLOTTING RESULTS

THE FOLLOWING MENU ALLOWS THE USER TO VIEW
AND SAVE 4 PREDEFINED PLOTS.

0 OVERVIEW PLOTTING FUNCTIONS

1 PLOT 1

2 PLOT 2

3 PLOT 3

4 PLOT 4

Choose an option: (ESC exits menu):

Figure 20. Plot Selection Menu

PLOT 1 GRAPHS THE FOLLOWING VARIABLES:

SCHCDT . . . . ESTIMATED SCHEDULE IN DAYS

PJBSZ . . . . PERCEIVED PROJECT SIZE IN TASKS

JBSZMD . . . . ESTIMATED PROJECT COST IN MAN-DAYS

TOTWF . . . . TOTAL WORKFORCE PEOPLE

CUMMD . . . . CUMULATIVE MAN-DAYS EXPENDED

PLOT 2 GRAPHS THE FOLLOWING VARIABLES:

CMTKDV . . . . CUMULATIVE TASKS DEVELOPED

CUMTKT . . . . .CUMULATIVE TASKS TESTED

CUMMD . . . . CUMULATIVE MAN-DAYS EXPENDED

PJBSZ . . . . PERCEIVED PROJECT SIZE IN TASKS

PDEVRC . . . . ESTIMATED % DEVELOPMENT COMPLETE

PLOT 3 GRAPHS THE FOLLOWING VARIABLES:

TOTWF . . . . TOTAL WORKFORCE

FRWFEX . . . . . . FRACTION OF WORKFORCE THAT IS EXPERIENCED

SDVPRD . . . . PRODUCTIVITY

COMMOH . . . . COMMUNICATION OVERHEAD

PLOT 4 GRAPHS THE FOLLOWING VARIABLES:

AFMPDJ . . . . ACTUAL FRACTION OF A MAN-DAY ON PROJECT

JBSZMD . . . . PERCEIVED TOTAL JOB SIZE IN MAN-DAYS

PJBSZ . . . . .PERCEIVED JOB SIZE IN TASKS

PMDSHR . . . . PERCEIVED SHORTAGE IN MAN-DAYS

Figure 21. Standard Plot Variables

PROJECT STATISTICS:

COMPLETION TIME	4.00	DAYS
TOTAL MAN-DAYS	3.35	MAN-DAYS
TOTAL DEV'T MD	3.35	MAN-DAYS
DESIGN & CODE	2.26	MAN-DAYS
QA MD	1.09	MAN-DAYS
TOTAL TESTING MD	.00	MAN-DAYS
OVERALL-PRODUCTIVITY	7,162.50	DSI/MAN-DAYS
TOTAL ERRORS	1.28	ERRORS
% ERRORS DETECTED BY QA	76.12	PERCENT

Figure 22. Simulation Results

```
                ***BAT.COM
                ***HELP.BAT
                *      *
                *      ***LIST.COM
                *            *          *PD.HP
                *            *          *DYNEX.HP
                *            *          *EDTR.HP
                *            **********SMLT.HP
                *                       *TOOLS.HP
                *                       *TRNS.HP
                *                       *UTILS.HP
                *                       *VIEW.HP
                ***VARIABL.BAT
                *       *
                *       ***DYNEX PROJECT.DNX
MENU.BAT**
                ***SMLT PROJECT *GO=
                ***VIEW PROJECT.RSL
                ***PLOT.BAT
                *       *               *REP PROJECT PLOT1.DRS
                *       **************REP PROJECT PLOT2.DRS
                *                       *REP PROJECT PLOT3.DRS
                *                       *REP PROJECT PLOT4.DRS
                *
                ***STORSTAT.BAT
                *       *
                *       ***REP PROJECT STATS
                *
                ***PD PROJECT.DYN
                        *
                        ***INSERT
```

Figure 23. File Hierarchy

K. PROFESSIONAL DYNAMO ENVIRONMENT

The user may work directly in the Professional Dynamo Environment by selecting option (8), Professional Dynamo Interface, from the main menu. Before the user is actually transferred to the Professional Dynamo Environment, a message is displayed stating that the user should only enter the Professional Dynamo Environment if he needs to change variables other than the seventeen accessible variables. The message also advises the user to read the Professional Dynamo user manual. The user at that point may hit "esc" to return to the main menu or hit "enter" to proceed to the Professional Dynamo Environment. The Professional Dynamo main menu is depicted in Figure 24.

Commands: Edit Compile Simulate View Tools Help Quit

Figure 24. Professional Dynamo Main Menu

Help on the Professional Dynamo options may be obtained through option (2), Help Facilities, of the Dynamica model main menu.

VI. CONCLUSIONS

A. ACCOMPLISHMENTS

1. User Interface

A literature search was conducted on user interface design. The principles and guidelines resulting from that research were applied in the design of the user interface. Significant areas of development included menu structure selection, standardizing menus, and improvements in control paths. Selection of the appropriate menu levels improved the performance and intelligibility of the interface. Previous non-standard portions of the interface were converted to consistent color and border schemes and menu layouts. Controls were added to change the menu paths to consistently place the user in an expected and easily recognizable location. A window directly to the model's environment was installed in the interface. Appropriate safeguards were included to adequately protect the user from inadvertently activating that window. Concentrated effort was expended to heighten the user's situational awareness by adding status information to verify the user's location, task status, and advice for what to do next.

2. User's Manual

A literature search was conducted on user's manual development. The guidelines and recommendations of that research were applied to the development of a user's manual.

B. LESSONS LEARNED

The earlier attempt at a user interface tried to fill retrieval times with messages and date/time stamps to assure the user the program was still functioning properly. The earlier version was developed on an IBM PC. Two years later, this effort was undertaken on an IBM AT compatible where the perceptible retrieval times had all but disappeared. The previous message fillers were now a waste of time and were therefore removed.

The previous design attempted to capitalize on modularity by creating separate batch files for virtually every task. Every message was separately contained in its own text file, presumably to facilitate future modifications. The present design consolidated many of the batch files which resulted in faster program execution and more flexibility in the menu structure. Many of the smaller messages, that typically were smaller than one screen, were incorporated into the body of the batch files which again shortens execution time.

C. FUTURE DIRECTION

The capability exists for on-screen help facilities, but only a token of this capability has been implemented. Extensive help facilities would enhance the educational value of the model.

The previous design saved program output by appending them to the same file. The present design provides the capability to identify a name for the file and a path to save the file under. If a file of the same name already exists, it will be erased and replaced by the new file. A safeguard should be programmed to warn the user if an old file exists before erasing it.

The model asks the user to accept the model's COCOMO values or to provide COCOMO values. COCOMO stands for Intermediate Constructive Cost Model (COCOMO), which is a type of cost-estimation tool for software development. A link to a COCOMO program at that point would be a valuable educational feature.

Presently the interface allows the user to change the values of the model variables. The user can verify what the values are by walking through the change process again. A strong recommendation would be to provide a single screen summary of the variables and their values that the user could view. Another recommended option is a display of the most recently changed variables and their values which the user could review before running the model.

The model provides the capability for a "gaming" feature. The gaming feature would be a natural option from the user interface and is therefore a recommended future enhancement.

LIST OF REFERENCES

1. Haury, Carson E., Design and Development of a User Interface for the Dynamica Model of Software Project Management, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1988.

2. Shneiderman, Ben, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Publishing Co., 1987.

3. Card, Stuart K., "User Perceptual Mechanisms in the Search of Computer Command Menus," Proceedings on Human Factors in Computer Systems, March 1982.

4. Covington, Michael A.,"Documentation That Works," PC Tech Journal, January 1985.

5. International Business Machines Corporation, IBM Research Report RC 9849, Doing the Same Work With Hardcopy and With Cathode Ray Tube (CRT) Terminals, by John Gould and Nancy Grischkowsky, 20 January 1983.

6. Hedin, Anne, "Unburden the User! Build and Buy Friendlier Documentation," Small Systems World, January 1985.

7. Pugh III, Alexander L., DYNAMO Users Manual, 6th Edition, Pugh-Roberts Associates, 1986.

8. Canova, Frank, Extended Batch Language Users Guide, Version 3, Seaware Corp, 1986.

9. Abdel-Hamid, Tarek k. and Madnick, Stuart E., "Managing Software Projects: An Integrative System Dynamics Modeling Approach," Naval Postgraduate School Working Paper No. 87-14, June 1987.

10. Abdel-Hamid, Tarek K. and Madnick, Stuart E., "An Integrative System Dynamics Perspective of Software Project Management: Arguments for an Alternative Research Paradigm," Naval Postgraduate School Working Paper No. 87-09, May 1987.

APPENDIX

BATCH AND TEXT FILE LISTINGS

```
:********************MAIN MENU BATCH FILE*****************
ECHO OFF
CLS
GRAPHICS
bat /N /p /s * ... Loading ... MODEL MENU
        Call -top1
        Exit
-top    goto -top%A
-top1 %A = 1
        color \1F
        ram
        cls
        begtype


                \1A                     MAIN MENU                    \1F


    \1A                 THE DYNAMICA MODEL OF SOFTWARE PROJECT
MANAGEMENT.         \1F


                        \1D 1   \1F MODEL REQUIREMENTS


                        \1D 2   \1F HELP FACILITIES
```

```
\1D 3   \1F SET MODEL VARIABLES

\1D 4   \1F RUN SIMULATIONS

\1D 5   \1F VIEW VARIABLE PLOTS

\1D 6   \1F VIEW STANDARD PLOTS

\1D 7   \1F STORE RESULTS

\1D 8   \1F PROFESSIONAL DYNAMO INTERFACE

\1D 9   \1F EXIT TO DOS


Choose an option:   (ESC exits menu):;
end
-1stkey1 inkey %0 | if %0 # = 1 type %0;
        if %0 = key01b return
        goto -%0~1
-2ndkey1 inkey %1 | if %1 # = 1 type %1;
        if %1 = key01b return
        if %1 = key020 goto -$%0$1
        if %1 = key00d goto -$%0$1
```

```
    if %1 = key008 goto -top1
    if %1 = key14b goto -top1
    goto -%0%11

-1~1 **** INTRODUCTION TO THE MODEL MENU ****
    CALL -top2
    bat /p /s goto -top1


-2~1 **** HELP FACILITIES ****
    BAT CLS
    BAT SHELL BAT /R HELP.BAT
    bat /p /s goto -top1

-3~1 **** VARIABLE MANIPULATION ****
  BAT CLS
    BAT SHELL  BAT /R VARIABL.BAT
  bat /p /s goto -top1

-4~1 **** RUNNING SIMULATIONS ****
  BAT CLS
    SMLT PROJECT -GO =
  bat /p /s goto -top1

-5~1  **** VIEW VARIABLE PLOTS ********************
    CALL -top3
    bat /p /s goto -top1

-6~1  **** VIEW STANDARD PLOTS      ************
    BAT CLS
    BAT SHELL BAT /R PLOT.BAT
```

```
        bat /p /s goto -top1

-7~1 **** STORING STATISTICS ****
          BAT CLS
          BAT SHELL BAT /R STORSTAT.BAT
          bat /p /s goto -top1

-8~1  **** PROFESSIONAL DYNAMO ENVIRONMENT ****
          CALL -top4
          bat /p /s goto -top1


-9~1 **** EXIT TO DOS ****
        BAT CLS
        BAT EXIT
        bat /p /s goto -top1

-%0~1
-$%0$1
-%0%11 beep goto -top1

-top2 %A = 2
          color \1F
          ram
          cls
          begtype
              \1A                DYNAMICA REQUIREMENTS                \1F

1. PROGRAM AUTHOR:
```

DYNAMICA IS A SOPHISTICATED SIMULATION MODEL FOR SOFTWARE
DEVELOPMENT PROJECT MANAGEMENT. THE MODEL WAS WRITTEN BY
DR. TAREK K. ABDEL-HAMID IN PROFESSIONAL DYNAMO.

2. HARDWARE REQUIREMENTS:

MEMORY: THE MODEL REQUIRES AN IBM OR IBM COMPATIBLE XT/AT MICROCOMPUTER,
WITH 640 K OF RAM AND A HARD DISK OR 1.2 MEG FLOPPY.
CO-PROCESSOR: ALTHOUGH NOT NECESSARY, THE USE OF A MATH CO-PROCESSOR
WILL DRAMATICALLY REDUCE THE SIMULATION EXECUTION TIME.
DISPLAYS: THE MODEL SUPPORTS MONOCHROME, MDA, CGA AND EGA GRAPHIC MODES.

3. SOFTWARE REQUIREMENTS:

THE MODEL REQUIRES MS OR PC DOS 2.xx OR GREATER.

\1A HIT ANY KEY TO CONTINUE \1F

END
INKEY
return
-%0~2
-$%0$2
-%0%12 beep goto -top2

```
-top3 %A = 3
        color \1F
        ram
        cls
        begtype


                YOUR RESULTS ARE BEING PREPARED FOR VIEWING !


                 IF YOU HAVE AN EGA GRAPHICS CARD TYPE (y)
                 FOLLOWED BY ENTER.

                 IF YOU DO NOT HAVE AN EGA CARD TYPE (n)
                 FOLLOWED BY ENTER.

END
BAT INKEY %0
BAT IF Y = %0 THEN GOTO -EGA
BAT ELSE GOTO -OTHER
BAT -EGA
VIEW PROJECT.RSL
BAT CLS
GOTO -ENDEGA
BAT -OTHER
VIEW PROJECT.RSL -PLM 6
BAT CLS
BAT -ENDEGA
return
-%0~3
```

```
-$%0$3
-%0%13 beep goto -top3

-top4 %A = 4
        color \1F
        ram
        cls
        begtype


        \1A                PROFESSIONAL DYNAMO ENVIRONMENT      \1F


        This option allows the user to bypass the basic Dynamica interface
        and work directly with the Professional Dynamo interface.

        The user should only select this option if there is a need to access more
        than the standard seventeen variables provided in the basic interface.

        The user should consult the Professional Dynamo reference manual
        before attempting to work from the Professional Dynamo interface.


      \1A       Press \1D<ENTER> \1Ato access the Professional Dynamo interface
\1F

                                   or

      \1A             Press \1D<ESC> \1Ato return to the Main Menu!     \1F
```

```
end
INKEY %1 | if %1 = key01b goto -top1
PD PROJECT.DYN
return
-%0~4
-$%0$4
-%0%14 beep goto -top4

-on.error-
 if %R > 82 if %R < 90 type !! Floating Point Error !! |goto -Calc.
 Cls beep type Unexpected batch file error %R in line %L |exit
```

```
:*********************HELP BATCH FILE**********************
BAT /P /S
        Call -top3
        Exit

-top3 %A = 3
        color \1F
        ram
        cls
        begtype




            \1A                   HELP FACILITIES                   \1F



                        \1D 1   \1F MODEL INTRODUCTION


                        \1D 2   \1F ONLINE HELP


                        \1D 3   \1F PROFESSIONAL DYNAMO HELP
```

```
Choose an option:    (ESC exits menu):;
end
-1stkey3 inkey %0 | if %0 # = 1 type %0;
         if %0 = key01b return
         goto -%0~3
-2ndkey3 inkey %1 | if %1 # = 1 type %1;
         if %1 = key01b return
         if %1 = key020 goto -$%0$3
         if %1 = key00d goto -$%0$3
         if %1 = key008 goto -top3
         if %1 = key14b goto -top3
         goto -%0%13

-1~3 **** DYNAMICA INTRODUCTION ****
       BAT CLS
BAT COLOR 1F
BAT BEGTYPE
             \1A              MODEL INTRODUCTION              \1F
```

The Dynamica Model of Software Project Management, created by Tarek K. Abdel-Hamid, is a comprehensive model of the software development process. The model, written in Professional Dynamo, integrates both management functions (e.g., planning, control, and staffing) with software production activities (e.g., design, coding, reviewing and testing).

A primary role of the Dynamica Model is to serve as an instructional aid to increase the manager's understanding of the software development process. By manipulating the program variables, the manager can see the effect changes have on the overall process.

The Dynamica Model can be an effective tool for the software project

manager in the actual management process. Variables, such as estimated project cost and schedule completion time, can be changed and simulations run within minutes. This rapid return of results, for contemplated program changes, allows the manager to consider more alternatives in greater depth.

A total of seventeen variables, from both the management functions and the software production activities, can be easily changed by the casual user. If a more knowledgable user has the requirement to alter the other Dynamica variables, a lower program level is accessable to make those changes.

\1A HIT ANY KEY TO CONTINUE! \1F

END

BAT INKEY

BAT CLS

bat /p /s goto -top3

-2~3 **** ONLINE MODEL HELP INFORMATION ****

BAT CLS

BAT COLOR 1F

BEGTYPE

\1A ONLINE HELP \1F

Online help is available from inside the following program sections:

SECTION NO.	SECTION NAME
3	Setting Model Variables

Access online help by typing 'H' when an input is required.

\1A HIT ANY KEY TO CONTINUE! \1F

END

INKEY

CLS

bat /p /s goto -top3

-3~3 **** PROFESSIONAL DYNAMO HELP ****

cls

begtype

\1A PROFESSIONAL DYANMO HELP \1F

Professional Dynamo help is intended for those users who

need to access directly the Professional Dynamo interface.

Option (8) of the main menu provides access to the Professional Dynamo interface.

Press <ENTER> to continue with Professional Dynamo Help

or

Press <ESC> to return to the HELP menu!

```
end
INKEY %3 | if %3 = key01b goto -top3


          call -top4
        bat /p /s goto -top3

-%0~3
-$%0$3
-%0%13 beep goto -top3
-top4 %A = 4
          color \1F
          ram
          cls
          begtype
```

\1A PROFESSIONAL DYNAMO HELP
\1F

\1A THE FOLLOWING MENU PROVIDES ACCESS TO PD HELP FACILITIES. \1F

\1D 1 \1F OVERVIEW OF PD HELP

\1D 2 \1F DYNEX HELP

\1D 3 \1F EDITOR HELP

\1D 4 \1F SIMULATION HELP

\1D 5 \1F TOOLS HELP

\1D 6 \1F TRANSLATOR HELP

\1D 7 \1F UTILITIES HELP

```
\1D 8    \1F VIEW HELP



Choose an option:    (ESC exits menu):;
end
-1stkey4 inkey %0 | if %0 # = 1 type %0;
        if %0 = key01b return
        goto -%0~4
-2ndkey4 inkey %1 | if %1 # = 1 type %1;
        if %1 = key01b return
        if %1 = key020 goto -$%0$4
        if %1 = key00d goto -$%0$4
        if %1 = key008 goto -top4
        if %1 = key14b goto -top4
        goto -%0%14

-1~4 **** OVERVIEW OF PD HELP ****
      BAT CLS
      L PD.HP
      bat /p /s goto -top4

-2~4 **** DYNEX HELP ****
      BAT CLS
      L DYNEX.HP
      bat /p /s goto -top4

-3~4 **** EDITOR HELP ****
      BAT CLS
      L EDTR.HP
      bat /p /s goto -top4
```

```
-4~4 **** SIMULATION HELP ****
     BAT CLS
     L SMLT.HP
     bat /p /s goto -top4

-5~4 **** TOOLS HELP ****
     BAT CLS
     L TOOLS.HP
     bat /p /s goto -top4

-6~4 **** TRANSLATOR HELP ****
     BAT CLS
     L TRNS.HP
     bat /p /s goto -top4

-7~4 **** UTILITIES HELP ****
     BAT CLS
     L UTILS.HP
     bat /p /s goto -top4

-8~4 **** VIEW HELP ****
     BAT CLS
     L VIEW.HP
     bat /p /s goto -top4

-%0~4
-$%0$4
-%0%14 beep goto -top4
```

:****************PROFESSIONAL DYNAMO HELP TEXT FILE*********

****

*   PROFESSIONAL DYNAMO PLUS HELP OVERVIEW   *

****

This is the Overview of the on-line Professional DYNAMO HELP facility. Each of the commands of PD Plus is described here. The HELP capability in PD Plus is "context - sensitive" so that when you call for help from a particular module, PD Plus knows where you are and will display the HELP screens appropriate to your location in the product.

You can call for HELP from PD Plus or from the command level of any of the following modules:

- Editor
- Simulator
- Viewer

- Translator

Each of the PD Plus modules listed above has its own Help files that provide more information about the particular module. With your cursor in the command line at the bottom of your screen, you may invoke HELP either by

Press Enter for the next page

typing the letter "H", or by moving your cursor over the word "Help" with the arrow keys and pressing the ENTER key.

If you desire more detail, please refer to the appropriate section in the PD Plus Reference Manual.

For more information on a particular module or PD Plus commands level, type the capilalized letter of one of the following:

- PD
- Edit
- Compile
- Simulate
-

View

- Document
- dyneX
- Report
- cOnvert
- reFormat
- Translate
- Quit and esc

Professional DYNAMO Plus Commands

The screen following the PD introductory screen is called the "top command level." This is the top of the command hierarchy. Your screen should be empty except for a highlighted line at the bottom with the following:

Commands: Edit Compile Simulate View Tools Help Quit

Each of these commands may be invoked by typing the first letter of the

command, or by positioning the cursor over the word with the arrow keys and pressing the ENTER key. (Note: the ENTER key will be used to refer to the carriage return.) A menu of all the files that might be processed by the command in the current directory are displayed for you to choose one by arrow keys and Enter. If PD was invoked with a path to some other directory, appropriate files in that directory will be displayed.

The EDITOR

Edit allows you to create and modify the various input files required by Professional DYNAMO:

.ASC - Source file for exogenous data
.DEF - Quantity definition file for Documentor (optional)
.DNX - Dynex screens
.DRS - Report
specification
.DYN - Original model

Edit has its own help file that can be displayed by typing ESC followed by the letter "H", or by pressing <F1>. If you type the letter "H", you will see a menu of topics for which Help is available.

Alternatively, you may press <F1> to go directly to the Help for the special function keys. This will display a "picture" of all the function keys; pressing any of these keys will result in an explanation of that key in the Editor.

The COMPILER

Compile translates your model (with the extension .DYN) for processing by Simulate. It creates three files, with the extensions .DAT, .INS, and .SMT. Should it discover any errors, you will be returned to Edit in the display error mode, where you can cycle through your errors one or more times by typing <Alt-N>.

A successful compile will terminate with a message indicating the size of your model in Bytes. You should be at PD top level with the cursor positioned on the Simulate command.

The SIMULATOR

Simulate simulates your model one or more times to produce .RSL files that can be formatted by View or Report according to your specifications. You may change the values of parameters and tables, change which variables will be saved in the .RSL file, run the model, preserve the final conditions, and resume those conditions as the initial conditions for additional run(s).

Simulate has its own help file for details on its use.

The VIEWER

View (and Report) display the run files produced by the Simulator in graphical or tabular format. Unlike Report, which requires a report specification file, View with a few interactive commands from the keyboard will display your results.

View has a Print option so that you may print hardcopy of your graphical or tabular results.

View has its own help file for details on its use.

REPORT GENERATOR

The Report Generator, REP, allows you practically complete control over the format of your output. You can compare values from different runs, compute the numbers to be displayed, choose the exact text to describe a number, and even specify how numbers are to be formatted. Graphical output can have the title of your choice, variables can be named as you choose, scales can be fixed, and the variables plotted computed by the Report Generator. To achieve all this you must create a Report Specification File (.DRS) with Edit.

See the help file under Tools for more information.

The DOCUMENTOR

The Documentor can document a model interspersing variable definitions between small blocks of equations, and creating alphabetic listings of all quantities with their definitions and where they are defined. It will also provide where-used lists in either of two formats.

See the help file under Tools for more information.

REFORMAT

Reformat "cleans up" a model by aligning equations and definitions in standard columns. Separate columns may be specified for "major" and "minor" equations and for definitions.

See the help file under Utilities for more information.

CONVERT

Convert will translate a model in Dynamo II or III format to Professional DYNAMO format.

See the help file under Utilities for more information.

DYNEX

Dynex is a model interface that allows a user with no knowledge of PD or the particular model to simulate it and view the results. Using Dynex the experienced model builder can make a model available for use in a structured and easily understood frame work. By responding to simple questions and prompts, an inexperienced user can make parameter changes, execute simulations and view the results.

See the help file under Tools for more information.

TRANSLATE

PD's exogenous variables facility requires that the values of these variables be supplied in a file (with extension .EXG) in a special format. Translate creates such a file from a spreadsheet (123, Symphony, or one you create with Edit) or from a PD simulation.

See the help file under Utilities for more information.

ESC and QUIT

The ESC command brings you up one level in the command structure. The ESC
command may be invoked not only by typing the letter "E" or by using the cursor, but also by pressing the ESC key on the keyboard.

Quit exits any PD program and returns you to DOS.

When you have invoked a module directly from DOS, both Quit

:******************DYNEX HELP TEXT FILE********************

End each line by pressing ENTER

Separate answers with space(s), comma, or slash --
so don't use the comma to group the 000's in large numbers.

The BACK SPACE key erases the previous character.

During any answer, you may invoke:

HELP (type h then press ENTER) gets you help, and returns youto the session
QUIT (type q then press ENTER) ends the session
RESTART (type r then press ENTER) restarts the session

:********************EDITOR HELP TEXT FILE***************

****
*   EDITOR ON-LINE HELP FACILITY   *
****

This help file explains the basic Editor commands and the special function keys. If you need more detail, please refer to Chapter 2 of the Professonal DYNAMO User Reference Manual.

For help on a specific area, select from the topics below by typing the first letter of the topic:

- Keys
- Editor commands (Save, Print, Return, Help, Esc, Quit)
- Functions supported by DYNAMO

Special Functions   Corrections and Cursor Movements

F1 F2   delete
help beeps   previous   Num Lock
[spl scrn] [2nd file]
char

F3 F4 top of up up
find find
file 1 line page
name literal Type F for < <page> < > <5 ln>

F5 replace F6 replace Dynamo Functions enter
left right
name literal 1 spce 5 1 spce
globally globally <1/3L> <1/3L>

F7 replace F8 replace end of down down
name literal * file 1 line page
verify verify Ctrl - < > <page> < > <5 ln>

F9 cut F10 reInsert Delete
text out paste deleted current
[line]
text back [ ] - Shift chars char

For more help, type any key you're interested in.

F1 provides help either
about the keyboard or the command that has been started.

F2 displays the reason for the last beep generated.

A "name" is a sequence of alphanumeric characters (letters and/or numbers) which is bounded by "delimiters" or non-alphanumeric characters (spaces, periods, slashes, parentheses, etc.). A "literal" is any sequence of characters, with or without delimiters.

F3 - Find name

You: F3

Computer: Find what?

You: <name> F3

A "literal" is any sequence of characters. In contrast, a "name" is a sequence of alphanumeric characters (letters and/or numbers) which is bounded by "delimiters" or non-alphanumeric characters (spaces, periods, slashes, parentheses, etc.).

F4 - Find literal

You: F4

Computer: Find what?

You: <literal> F4

A "name" is a sequence of alphanumeric characters (letters and/or numbers) which is bounded by "delimiters" or non-alphanumeric characters (spaces, periods, slashes, parentheses, etc.). A "literal" is any sequence of characters, with or without delimiters.

F5 - Replace name from cursor on

You: F5

Computer: Find what?

You: <current name> F5

Computer: Replace with?

You: <desired name> F5

A "literal" is any sequence of characters. In contrast, a "name" is a sequence of alphanumeric characters (letters and/or numbers) which is bounded by "delimiters" or non-alphanumeric characters (spaces, periods, slashes, parentheses, etc.).

F6 - Replace literal from cursor on

You: F6

Computer: Find what?

You: <current literal> F6

Computer: Replace with?

You: <desired literal> F6

A "name" is a sequence of alphanumeric characters (letters and/or numbers) which is bounded by "delimiters" or non-alphanumeric characters (spaces, periods, slashes,parentheses, etc.). A "literal" is any sequence of characters, with or without delimiters.

F7 - Replace name from cursor on
if you type y(es)

You: F7

Computer: Find what?

You: <current name> F7

Computer: Replace with?

You: <desired name> F7

Computer: Replace ? (Y or N)

A "literal" is any sequence of characters. In contrast, a "name" is a sequence of alphanumeric characters (letters and/or numbers) which is bounded by "delimiters" or non-alphanumeric characters (spaces, periods, slashes, parentheses, etc.).

F8 - Replace
literal from cursor on if you type y(es)

You: F8

Computer: Find what?

You: <current literal> F8

Computer: Replace with?

You: <desired literal> F8

Computer: Replace ? (Y or N)

F9 - Cut text out and place in buffer

You: F9

Computer: * * * Cutting * * *

You: Down- and/or Right-arrows

(or

Up- or Left-arrows)

Computer: highlights text to be cut

You: F9

F10 - Insert contents of buffer following cursor

You: F10

Computer: reinserts text that was last

cut out with F9 key

<Shift F1> - Split screen/Jump to other screen

Typing <Shift F1> jumps you between the upper and lower screens or splits

the screen to give you two screens which can be moved independently in one

file.

For a detailed explanation of <Shift F1>, see section 2.2.5 of your PD Reference Manual.

<Shift F2> - Read new file into split screen

Typing <Shift F2> allows you to specify the name of a file to be read in place of the file just being processed. If only one file being processed, it is retained for two file processing. (The new file is always located in the lower screen.) Text may be cut from one screen and pasted into the other.

If you type Esc instead of a new file name, split screen operation is cancelled. (If you are processing two files, the file in the screen where the cursor was when <Shift F2> was typed, is discarded.)

For a detailed explanation of <Shift F2>, see section 2.2.5 of your PD Reference Manual.

The white keys in the middle of the keyboard are the standard input

characters: what you type is what you get on the screen.

To get special characters such as * and (, you must hold down one of the SHIFT keys while you type the appropriate key. The SHIFT keys are in the second row of the keyboard, marked with large, open up-arrows.

?

SHIFT SHIFT

/

Space Bar Caps

Lock

The white keys in the right-hand keypad may be used either to move the cursor or to insert digits (and decimal points). This depends on the SHIFT, CTRL, and NUM LOCK

delete
previous Num Lock
char

top of up up

SHIFT KEY file 1 line page

up down < <page> < > <5 ln>

NUM LOCK on numbers arrows enter left
right
1 spce 1 spce
off arrows
numbers <1/3L> <1/3L>

end of down down
SHIFT * file 1 line page
For a more detailed explanation, see <page> < > <5 ln>
section 2.2.2 of your Professional

DYNAMO Reference Manual.

reInsert Delete
Caps deleted current
For more help, type any key you're Lock chars char
interested in.

Professional DYNAMO Editor Command Line

When you invoke the Editor you will see its command line in inverse video at the bottom of your screen. The commands are:

Save Print
Return Help Esc Quit

To invoke any of these commands type the ESC key to position your cursor in the command line and either type the first letter of the command or type

arrow keys to position the cursor under the desired command and Enter.

The Save command will save your file exactly as you see it on the screen. When you "save", PD asks you to:

Type new name or type Enter to save to: <original name>

You may type Enter to save your file under its orginal name, type a new name

and Enter, or type Esc if you do not wish to save your file at this time.

page 1 of 2

If a file exists with the same name, its extension will be renamed .BAK; if a file also exists with the same filename and extension .BAK if will be deleted unless it is read only.

When PD completes a Save, your cursor is positioned under the command "Return." If you type ENTER (or type "R"), PD returns you to editing your file at the location you were just before pressing the ESC key.

The Print command creates a simple listing your file on your printer. (See Reformat for listings with page headings.) Caution: if your printer is not ready, your computer will lock up.

If you invoke the Esc command, PD leaves the Editor and returns to the PD top command level. If you have made modifications to your file since your last Save, PD first asks if you wish to save your file before leaving the Editor. You may respond by typing "Y" for yes, "N" for no,
or Esc to remain in the Editor.

Quit works similar to Esc except you are returned to DOS rather than PD

The following functions are available in DYNAMO:

Built-in Math Functions

COS(a) LOGN(a)

SIN(a)

SQRT(a)

EXP(a)

Built-in PD Functions

CLIP(p,q,r,s) NOISE() STEP(height,steptm)

DELAY1(in,del) NORMRN(mean,std_dv) SUM(array)

DELAY3(in,del) PRDV(vctr,frst,last)

SUMV(vctr,frst,last)

DELAYP(in,del,ppl) PULSE(hght,wdth,frst,intvl) SWITCH(p,q,r)

DLINF3(in,del) RAMP(slope,start) TABLE(tab,x,xlow,xhigh,xincr)

FIFGE(p,q,r,s) SAMPLE(x,intvl,isam) TABHL(tab,x,xlow,xhigh,xincr

:*******************SIMULATOR HELP TEXT FILE***************

Simulator Help Facility

The Simulator will execute a selected model over the time period you choose. It requires as its input a compiled model (three files with the name of your model followed with the extensions .DAT, .INS, and .SMT).

When you invoke the Simulator from PD top command level, it will display a menu listing all files with the .SMT extension and you will be prompted to select one of these. However, if the Simulator locates only one such file, it will select this file and proceed with the next screen.

After you have chosen the model to be simulated, you will be in the Simulate command line with the following commands available:

- Changes
- Save
- Go
- Preserve
- Resume
- Esc, Quit (type "E")

For more information on a command, type the first letter of the name or type "K" for help on Runge-Kutta integration).

Changes

The Changes command allows you to alter the input values of any of your given constants, tables, or specification information. Invoking this command will cause a full-screen display of all the inputs, grouped by type. The arrow keys will move you from number to number and new values can be typed in directly over the old ones.

When you have finished modifying your constants and table values, press the ESC key to return to the Simulate command line.

In the case of tables the cursor can also be moved to table names as well as to their values. If you press the ENTER key with the cursor over a table name, you will be presented with a graphical plot of the table. To modify this curve, use the right (or left) arrow keys to move to the location of the point you wish to modify. Notice that when you first pressed Enter, your cursor (marked as an "X") was positioned over the left-most point on the graph.

Press Enter for the next page

When you have moved to the point you wish to change, use the up (or down) arrow keys to move the point. When you change the position of the point, the original curve is displayed as a dashed line.

After you are satisfied with the position, you may either move to the next point you wish to modify or quit by pressing the ESC key. Pressing the ESC key will put you back into Changes mode, with your cursor on the table name that you just modified; any changes that you made will be reflected in the actual values in the table.

Should you wish to alter the value in a table numerically, you may do so as well.

Save

The Save command allows you to select the variables whose values will be saved for later viewing. Invoking Save, will cause a display to appear that lists all variables (grouped by type); those that have been previously saved will be highlighted. The arrow keys will move the cursor between variables and pressing the ENTER key on a variable will allow you to toggle between save and un-save.

ESC will return you to Simulator command level.

Go

The Go command invokes the execution of the model run. The Simulator will save the results of the run under the file name "model name.RSL", unless you specify another name. If you make several runs and use the same filename for saving the results, each successive run will write over the results from the previous run.

Should you wish to terminate a simulation in the middle of an execution, press the ESC key. The next time Simulate writes results to the disk you will be returned to Simulator command level.

Once a simulation is finished, you will be back in the Simulator command line and you may make changes (or alter saved variables) and execute another run, or ESC back to PD top command level to invoke the View module. Alternatively, you may wish to "preserve" the state of the model from the last run and "resume" the run over an additional time period.

Type Enter for Runge-Kutta integration

Runge-Kutta integration

To use the variable step size third order Runge-Kutta integration method set REL_ERR to a non-zero value (e.g. .01). REL_ERR is the relative error tolerance, the tolerable error relative to the current value of the level. Should the level become very small, the absolute error tolerance, ABS_ERR, will come into play. This tolerance is applied to all levels; it should be chosen considering which levels might approach zero and what error is tolerable in them.

If you are using this method and NOISE or NORMRN you must also set DT to some
value over which the value of these functions may be held constant. The choice should be made carefully, as a small value may increase the time required to carry out a simulation. See the Reference Manual for more details about this integration method.

Preserve

The Preserve and Resume commands allow you to make a short run, preserve the conditions that exist at the end of that run, and then make a series of runs starting with the conditions that were preserved. This is especially useful when running a model up to the present time and then making a series of exploratory runs over future time.

After making the initial run, you may wish to make a change or two to your parameters before preserving them. For example, changing LENGTH before preserving the conditions will save you changing it every time you resume the model.

When you issue the Preserve command, The Simulator will request the name of a

"state file." This is the file that will contain the current "state" of the model (the final values of all variables and the current values of all constants and tables). Either press Enter to use the default name, "model name.STT", or type the name of your choice. Simulate will add the extension ".STT".

Type Enter for Resume

Resume

Resume Invoking Resume will cause a menu of state files to be displayed, for you to select the one you want (unless there is only one). Once the selection has been made you are in the resume mode; every run you make will be made with the conditions found in that file, unless you issue Resume again to designate another file or leave Simulate.

If you wish to alter one or more computed constants or reinitialize a level, you can do so by computing them with K rather than N equations. See Chapter 3 of the Reference Manual for more about N and K equations.

Esc

The Esc command will bring you up one level in the command hierarchy. If you are making changes to variable values or save selections, invoking Esc will bring you back to the Simulate command line. A second invocation will take you out of the Simulator and bring you up to PD top command level.

Quit

The Quit command brings you directly to DOS, by-passing any other command level in Professional DYNAMO.

:*****************TOOLS HELP TEXT FILE*********************

Tools Help

Tools consist of four commands:

Document

dyneX (type "X")

Report

Utilities

Type the appropriate letter for help with that command.

Document

The Document command will provide you with various "reports" on your model, which include a listing of your model with definitions following every block of equations, a listing of all your variables (alphabetically sorted) and their types. Lists of where variables are used are provided in two formats. Invoking this command will cause a menu of all the available options to appear. The options are Boolean or numerical and the defaults are filled in. If you run the Documetor with no changes to the defaults, you will get all of the reports; you may suppress as many of these reports as you like by changing the appropriate value. Two of the fields in the Document options require a file name as a value, one of which is left empty as a default.

The last option in the list concerns the output. As a default, Documentor will automatically write the documentor listing into a file called "model_name.DOC". You have three alternatives to this: you can direct the output to screen by typing "CON" in the name field; you can type "PRN" to direct the output to a printer; or you can type a file name of your choice in the field.

page 1 of 2

The other option is concerned with a "definition file". This is an optional file in which variables and their definitions are stored separately from the actual model itself. Specifying a definition file causes the Documentor to look for definitions in the model and in the separate file. Definition files must carry the extension ".DEF." The default value for this option is a blank, indicating that no definition file is to be used. If you wish to change this, you must type in the name of the definition file in the appropriate option field.

When you have finished modifying the options, you must signal the Documentor to begin by pressing the Esc key.

For more more information see chapter 11 of The Professional DYNAMO Plus Reference Manual.

DYNEX HELP FACILITY

To use DYNEX the model builder constructs and debugs a model, and then designs a sequence of informational screens and user inquiries to guide an inexperienced user in the choice of parameters and tables to make a simulation experiment.

The user (with the help of a DOS .BAT file) invokes DYNEX to help decide the parameter changes and output specification, SMLT to run the simulation (without further user intervention), and REP to report his or her results. This sequence may be repeated to make additional simulations.

Supposing that our model is called POP.DYN, the command "DYNEX pop" can be issued after POP.DYN has been compiled to give POP.SMT and POP.DAT, and after the builder has used EDIT to create the ".DNX" file POP.DNX, containing explanations and queries for the user.

DYNEX reads POP.SMT and POP.DAT, to create a table of default values for all parameters in the model. Then it interactively reads POP.DNX, displays explanations and prompts on the end-user's screen, and elicits the user's values and choices. The values are written into POP.DAT; text output is written to POP.DRS (to be used by REPort). After the user finishes DYNEX, he or she (or the builder's ".BAT" file) activates SMLT, to simulate the new POP.DAT, giving POP.RSL. REP can report upon POP.RSL, using the PLOT and REPORT statements chosen by the user, and copied into POP.DRS.

Options allowed are:

DYNEX pop -d outfile : redirect text output to outfile.drs
DYNEX pop -h hlpfile : read helpful information for the user from hlpfile.hlp
DYNEX pop -l : logically list pop.dnx, showing internal structure

The statements understood by DYNEX are :

DISPLAY : each subsequent line is displayed to the end-user, until DEND.

DISPLAY CLEAR: clear the screen before displaying.

DEND : stop displaying.

DENDQ : stop displaying, and turn off DYNEX's next prompt to the end-user.

CQ, TQ, BQ, IQ, PQ, or SPECQ

: read the parameter's name; display its values; elicit new values from the user.

CHOICE n : there follow n sections of statements. Ask the user to choose one. Process all statements in that section; ignore the other sections.

CHOICES n : there follow n sections of statements. Ask the user to choose one or more. Process all statements in those section; ignore the other sections.

CEND : ends one of n sections.

DYNEX -d outfile : redirect text output to outfile.drs.

DYNEX -h hlpfile : read helpful information for the user from hlpfile.hlp.

DYNEX -l : logically list the ".dnx" file.

C, T, B, I, P, or SPEC

: read the parameter's name; copy the values on the statement to the ".DAT" file.

INSERT name : process the contents of the file name.

other statements

: copy these text statements to POP.DRS (or the file specified by the most recent DYNEX statement).

This is the last screen for DYNEX

REPORT GENERATOR HELP

THE PD Plus Report Generator (REP) allows you to format simulation results into easily comprehensible graphs or reports. Graphs can be the traditional PLOT of variables over TIME, or of one variable versus another (PLOTXY). Reports can be specified down to the last comma and column position.

To use the PD Plus Report Generator you first use the Editor to create a "Report Specification" file, then use Simulate to make one or more simulations called for by the specifications, and finally issue the Report command to generate plots and/or reports.

For more Help on any topic type the first letter of the following:

- Starting the Report Generator
- Plot statements
- Tabular output
- Glossary

STARTING the Report Generator from DOS

The command REP has two arguments, as in:

REP POP1 POPPLT

The first argument (POP1) is the name of the result file (.RSL) created when you executed the Simultator using POP1. The second argument (POPPLT) is the name of a Report Specification (.DRS) file which you create with Edit using the Report Generator language described here. If the .RSL file and .DRS

file have the same name the second may be omitted.

There are six options you can specify:

-outf file   write all output to screen & all text output to file
(instead of to pop1.out in the example above)

-plm #   PLot Mode 6 (CGA), 14 (EGA 200x640), 16 (EGA 350x640),
or
40 (bw Hercules)

-plpr   PLot with PRinter characters

-plw #   PLot Width in characters = #

-t   rewrite output file if it already exists

-txi #   Time (or X) value every # SAVPERs

PLOT statements

You specify plotted (either true graphic or "printer-plotted") output by listing the variable names on a PLOT statement in your .DRS file:

PLOT TTPOP, <POP>

XY plots are permitted; the first variable named on the PLOTXY statement is the independent variable; the following variables are the dependent.

PLOTXY UNEMP, INFL

<SCALES>

To force a group of variables onto the same scaling group, surround the group with <...>. To specify your choice of scales, include your lower_scale and upper_scale just before the >.

<A,B,C,0.,4>

~LINE_STYLE

You can specify the line_style by following the variable name with ~ and a digit (broken lines), capital letter (colored lines) or small letter (colored and broken lines).

(SUBSCRIPT)

Arrays in PLOT statements can be unsubscripted (plot all elements) or can have PD Plus subscripts: literal constant - POP(1); element name - POP(YOUTH); FOR loop name - FOR AGE = YOUTH to ELD do PLOT POP(AGE) END; or the dummy subscript - POP(*). REP creates a curve for each element, for up to 6 curves.

.RUN

REP can produce plots showing results from several different simulations. The default result file is the one you specify as the first argument to REP. To plot a variable from a result file other than the default one, you follow the variable name by a "runscript", e.g., .POL2. (To change the default assign a new name to RUN (RUN = BASE).)

PLOT ABC, ABC.POL2

[TIME]

If you may want to freeze time for a variable on a PLOT statement, you can follow a it by a timescript, [time]. For example, to normalize a variable, you may divide by its value at TIME = 0.

PLOT ABC/ABC[0]

"TITLE"

You can give a graph a title by inserting the desired text (up to 60

characters between double quotes) after PLOT, but before the usual plotting information.

For example:

PLOT "FINAL SECTOR" IAR

To give your own name to a curve, put the text in double quotes after the variable name. For example:

PLOT "FINAL SECTOR" IAR "INVENTORY", RRR "ORDERS"

This is the last page for PLOT

Tabular output

To create a Report Specification you first conceive of your report as a number of lines of printed material, each line made up of one or more fields of information. A field starts at, ends at, or is centered at some column. A field can contain simple text such as "Net profits", a value saved in a .RSL file, or the result of an arithmetic expression computed by the report generator (generally from values in the .RSL files).

A report specification is a sequence of field specifications that tell what is to be placed in the next field or modifies the environment generating the report. The environment is the current values of all the "local variables" (mostly used is subscripts), "text labels", and "run-ids" which in addition to the "saved quantities" make up the variables of the report generator.

A field specification may be a saved quantity, an expression, a local

variable, a text string, or a text label, all of which are displayed. Additionally, a field specification may be an assignment to a local variable or a text label, "NEWPAGE", or a control block. A control block is

FOR ... DO ... END,
HEADING ... END,
IF ... THEN ... END,
IF ... THEN ... ELSE ... END.

Fields are separated by commas (or semicolons to also indicate that a new line should be started). No delimiter is required before a reserved word or text string.

All NAMES, whether local variable, run id, saved quantity, or text label, are formed by the same rules as DYNAMO.

A LOCAL VARIABLE is a name that is not one of the other types of variables.
(TIME is treated as a local variable even though it can be found in the .RSL file. Also all element names are local variables with appropriate values.)

A RUN-ID is recognized by context, but there also must be a .RSL file by that name.

A TEXT LABEL is recognized by assigning it to a text string or a vector of text strings (separated by "/"s). For example:

RGNNAM = "NORTH"/"SOUTH"

To specify an element of this vector add a subscript to the name:

RGNNAM(i)

A TEXT STRING is formed by enclosing text between single or double quotes

('...' or "..."). To minimize the number of error messages produced when REP finds a text string with no closing quote, you must specify the entire text string on one single line.

Two special text labels that define the position of fields in a line and the format of numbers are FORMAT and PICTURE.

FORMAT specifies one column of a field and whether it is first, last, or middle. The FORMAT consists of pairs of column numbers and justification characters. The valid pairs are:

nn <	left justify, starting in column nn
nn >	right justify, ending in column nn
nn -	center on column nn

An example is REP's default (spaces are ignored):

FORMAT = "1< 41<" ,

PICTURE specifies how REP prints a value. The characters in the picture specify where digits are to be printed, whether a leading zero should be printed, how and where a negative value should be indicated, and the positions of visible and implicit (used when converting to decimal) decimal points.

The characters recognized include:

9	A digit, even a leading zero, is printed in this column.
Z or z	A digit, but not a leading zero, is printed in this column.
- (or)	is printed when the value is negative.
db or cr	in either case is printed when the value is negative.
V or v	indicates the position of an implicit decimal point, but does not cause the printing of any character.

. , or / is printed when it follows a printed digit or "V".

$ is printed.

Multiple "-", "(", or "$" indicate that the character should be printed directly before the leading non-zero digit when appropriate.

PICTURE, like FORMAT, can be reassigned as often as required. REP starts with the default assignment of PICTURE = "---,--9v.99".

The results of ARITHMETIC EXPRESSIONS may be displayed or assigned to a

local variable. Expressions may involve SAVED QUANTITIES, LOCAL VARIABLES,

and/or NUMBERS. For example:

(POP(AGE)-POP(AGE).BASE)/POP(AGE).BASE

PAGENO = PAGENO+1

The default values of TIME and RUN determine the particular value of a saved variable used in an expression, unless [...] follows it to specify a different value of TIME or .RUN-ID follows it to designate a different run. The order of the local TIME and RUN does not matter. The default values can be changed in an assignment statement. For example:

RUN = BASE, TIME = 83,

CONTROL BLOCKS simplify the construction of large reports. Control blocks may be nested to practically any depth. Indenting lines to make it easier to match END with its head is advised (blanks within a report specification have no meaning except in text).

FOR ... = ... TO ... DO ... END

FOR ... = ... TO ... BY ... DO ... END

FOR loops let you repeat a part of your report specification with TIME or a subscript modified. For example, to produce the same report for TIME equal to 80, 85, 90, 95, and 100 you can write:

```
FOR TIME = 80 TO 100 BY 5 DO
    ...
    END
```

If the step size is 1 you may omit "BY 1".

FOR RUN = ... DO ... END

FOR loops can also be constructed to repeat something for different default RUNs. The several run-ids are separated by commas. For example:

```
FOR RUN = pol1 ,pop2, pol3 DO
    PLOT ttpop/ttpop.base
    END
```

HEADING ... END

You can specify what you want for the top of each page by including one or more HEADING procedures. For example:

```
PAGENO = 0,
HEADING
    FORMAT = "38- 75< 80<",
    PICTURE = "Z9",
    PAGENO = PAGENO+1,
    "TITLE", "PAGE", PAGENO;;;
    END,
```

Head ... end does not cause a heading to be printed; the current heading is activated by the reserved word NEWPAGE.

IF ... THEN ... END

IF ... THEN ... ELSE ... END

You can include, exclude, or choose among fields based on the comparison of two values (either or both of which may be expressions). For example:

```
IF SCORE[90] > SCORE.BASE[90] THEN
    "This policy is superior!"
ELSE
    "This policy is inferior!"
END
```

When the report specification follows one or more PLOT statements, the keyword REPORT is required.

This is the last page for Tabular output.

GLOSSARY

The DELIMITERS are:

- , separates fields
- ; separates lines
- = assigns value to a local variable, TIME, RUN, or a text label.
- () surrounds a subscript
- [] surrounds a local value of TIME
- . precedes a local value of RUN (or denotes a decimal point)
- ' ' surrounds a text
- " " surrounds a text

/ separates texts in a text label vector (or means "divided by")

< > surrounds a group of saved quantities to be plotted on the same scale

~ precedes a line_style

* stands for all elements in one dimension of a PLOTted saved quantity

+ - * / ( ) to form an arithmetic expression of saved quantities and/or local variables

< means "is less than"

> means "is greater than"

Press Enter for Reserved Words

RESERVED WORDS

BY	FOR	NEWPAGE	PLOTXY	THEN
DO	FORMAT	PICTURE	REPORT	TIME
ELSE	HEADING	PLOT	RUN	TO
END	IF			

Utilities' help file can be reached from within Utilities.

;*******************TRANSLATOR HELP TEXT FILE**************

Translator Help File

PD Plus allows you to specify a variable as having exogenous values, that is, values generated outside the model. To simulate a model with exogenous variables you must supply values for those variables over the course of the simulation. The Translator (TRNS) allows you to use data stored in spreadsheet and standard text files as the input to exogenous variables. The Translator supports spreadsheet files generated using Lotus 123 or Symphony, standard text files (also called ascii files) generated by the PD editor, and result format files generated by the Simulator or the Translator.

Type the initial letter (or digit) for more information on:

- 123 and Symphony spreadsheet files
- Ascii (text) files
- PD generated (.RSL and .EXG) files
- translate Commands (type "C" for general discussion)
- Load
- Review
- Select
- Write

123 and Symphony spreadsheet files

The translator needs to know for what time each value applies. This is done by putting the word TIME followed by numbers representing the times. TIME can be put at the beginning of a row and increase along the row or at the head of a column and increase along the column. In either case the increment in time from one cell to the next must be constant, though a value need not be filled in for every cell. More thanone TIME statement can occur: A TIME statement determines the times for the variables that follow

until another TIME statement is encountered.

Variables are just like time; put the variable name at the head of a row or column and follow it by values. The column (row) a value is in determines the time it is for. When values are not available a cell can be left blank, or the letters NA can be filled in. The Translator will load the result of spreadsheet formulas as they appear when the spreadsheet was saved.

Ascii (text) files

Standard text files are files containing only standard keyboard characters; such files are also referred to as ascii files. A standard text file can be created using many editors including the Professional DYNAMO editor. The required format for a text file is similar to that of a spreadsheet file.

Again every value needs a time for which it applies. The times are specified by the keyword TIME followed by the times. The times must be increasing by a constant amount. Unlike a spreadsheet file the TIME can only run horizontally.

Variables are specified by their name followed by their values. The times for the values start at the first value specified on the TIME statement, and increase by the increment between the different times. When the number of values is different from the number of times specified the times are extrapolated. Unlike a spreadsheet file horizontal spacing makes no difference. New lines are ignored so that values can be put on as many lines as is convenient.

PD generated (.RSL and .EXG) files

Result format files are output of the Simulator and the Translator. The Simulator creates a file with extension .RSL, the Translator a file with extension .EXG.

Result format files are set up by DYNAMO in a specific manner. Though you cannot control the way result files look, you can control what is in them. Results from a simulation are stored for saved variables, and no other variables in a model. Thus, to use a simulated value for a variable you must insure that the variable was saved during simulation.

Translator Commands

The Translator generates exogenous values for a given model. The process requires that files containing the data first be loaded using the Load command. Any number of files may be loaded, though one will often suffice. After the files have been loaded you may review the data that has been loaded using the Review command, the default command when all of the required variables have not been found after a Load. If more than one of the files loaded contains a given variable, you may select which set of values you wish to use, using the Select command. Finally, when you have loaded the desired values, you can create the exogenous data file with the Write command.

The command line in the Translator works like that of the other modules. Type the first letter of the command, or position the cursor over the command and press Enter.

Enter the first letter for more information on the individual commands:

Load

Review

Select

Write

Load

When you select Load the names of all files in the current directory that are available for loading will be displayed. The types of files displayed are .WKS for Lotus 123 files, .WRK for Lotus Symphony files, .ASC for standard text (ascii) files, .RSL for Simulator output files and .EXG for Translator output files. You choose the file you want to load by moving the cursor to it and pressing Enter, Esc will return you to the menu without loading any file.

Once you have chosen a file you will be asked to verify its attributes. Spreadsheet files are assumed to have time running across the columns unless you specify otherwise at this point. If there is no TIME statement in a spreadsheet or standard text file you may insert values for the starting time and the increment at this point as well. If a TIME statement is encountered these values will be ignored. Finally you may choose to use the file or not. If the file is not used then it will be ignored, or unloaded if it has already been loaded. Pressing Esc completes the loading.

If Load gets values for all the required exogenous variables the Write prompt will be highlighted, otherwise the Review prompt will be highlighted.

Review

The Review command causes a screen to appear that lists all of the files that have been Loaded, and all of the exogenous variables for which values

are needed. When no values have been found for an exogenous variable the variable name will appear highlighted. Otherwise, the time range over which those values applies will be given in parenthesis after the variable name.

You may move about the Review screen, but you cannot change any attributes. Pressing Esc will return you to the Translator menu.

Select

When values for an exogenous variable have been found in more than one loaded file it is necessary to choose between the different values. By default the Translator will use the value from the first file Loaded; Select allows you to change this.

When you choose Select a list of all variables that are contained in more than one source file is shown. For each of these variables the different source files as well as the starting and ending times are listed. The highlighted file shown is the file that will be used. By moving the cursor to a different file and pressing Enter the new file will be selected. Pressing Esc returns you to the Translator menu.

Note that you can deselect all the variables in a file by using the Load command and highlighting the Not Used cell.

Write

The Write command writes all the data that has been Loaded and Selected.

When the write command is issued you will be asked to give a file name to write to. The default file name is the model name and will usually be appropriate. If you choose a different name you will have to use an option when invoking SMLT. Pressing Enter causes the default name to be used.

After you have supplied a name the data is reorganized and written into the specified file. Any gaps in the data are filled in by linear interpolation, and missing values outside the available data range use the first or last value as appropriate.

:******************UTILITIES HELP TEXT FILE**************

Utilities Help

Utilities consist of three commands:

Convert

Reformat

Translate

Type the capiltalized letter for help on that command.

Convert

The Convert command has been provided so that users who have created a model with an older version of DYNAMO (II or III) can run the model with Professional DYNAMO. Because there are differences between PD and other DYNAMO versions, you must convert the older models using the Convert command.

The only option available concerns the name of output file. As a default, the Converter will write the converted model to disk in a file named "model name.DOC". If you wish to change this, you may type in a new model name.

Reformat

The Reformat command is used to "clean up" a model so that it is easier to read and explain. Invoking the command will cause a screen of options to appear on the top part of the screen. Each option has either a "Y" or "N" or a number as a default value. You may alter these defaults by using the arrow keys to move from field to field.

The last option in the list concerns the output. As a default, Reformat will write the reformatted model into a file on disk called "model name.DOC". You have three alternatives to this: you can direct the output to screen by typing "CON" in the name field; you can type a file name of your choice in the field; or, you can type "PRN" to direct the output to a printer. If you choose to direct the output to another file name, Reformat will write the output into a file with the name EXACTLY as you have typed it.

:*******************VIEWER HELP TEXT FILE****************

Viewer Help Facility

The Viewer reads the .RSL files generated by the Simulator and displays results in either grpahical or tabular format. You may specify a number of different views and return to an earlier one by simply typing its number.

When you invoke the Viewer, it will display a menu of all files with the extension .RSL for you to select one. If there is only one such file, the Viewer will use this file and proceed with the next screen.

View has five commands in addition to Help; for more Help on any one type the first letter of the command.

- Plot
- Select_Tabulate
- Tabulate_All
- ESC and Quit (type "E")

Plot

This command will generate plots of the variables you select. When you invoke it, you will see a menu of the names of all the variables that you have saved. You can select the variables by positioning your cursor over them and pressing the ENTER key; a selected variable will appear highlighted. After completing your selection, press the ESC key to plot your variables on your screen.

If you have a graphics monitor and card, the plot will be high- resolution bit-mapped graphics; if you have no graphics capabilities, you will see a character plot.

As a default, all the variables that you have selected will appear on separate vertical scales. If you wish several variables to share a common scale, position your cursor over the first variable and type "<" or ",". Select additional variables to share this scale with the ENTER key until the last, which should be selected with ">" or "." key. You may continue to select additional variables to be plotted on the same graph, either with separate scales or with a different common scale.

Type Enter for the next page

Under the graph will appear the following prompt line, which will remind you what can be done next.

View #:   Next   view_no   Print   Esc   Quit

View # is the number of this view, which may be entered later to return to this plot.

Pressing N for Next or ENTER will advance you to the next view. If it does not yet exist you will be given the opportunity to create it.

Typing the number of an existing view, and then pressing ENTER, will return that view to the screen.

If you have the proper printer and have issued the proper preparatory DOS commands, you may print the graph by pressing P. See Chapter 16 of the Reference Manual for details.

ESC and Quit return you to View and DOS respectively.

Select_Tabulate:

This command is similar to Tabulate_All except that you may choose which of your saved variables you want tabulated and in what order. When you invoke this command, you are presented with a menu listing all the variable names that you have saved.

To select a variable, position the cursor over it with the arrow keys and press the ENTER key; the name will appear highlighted. When you have completed your selections, press the ESC key. You will be queried for a print interval, the TIME interval between values. Either select the default by pressing Enter, or type another value. The table will appear.

To move around your table and view information that is after column 80 or below line 24, you may use the arrow key pad. The arrow keys will move you one row (column) up or down (left or right). To move by pages, use the Pg_Up or Pg_Dn keys. To scroll 7 columns right or left, press the <Ctrl> key together with the right or left arrow key. To move immediately to the top or bottom of the table, use the Home or End key, respectively.

page 1 of 3

At the bottom of the screen is a reminder of the options you now have:

View #: Arrows Home End <ctl>^Q <ctl>^P PgUp PgDn Next view_no Print Esc Quit

View # is the number of this view, which may be entered later to return to this view.

The Arrows will move you one column or row at a time. <ctl>^Q, <ctl>^P, PgUp,
and PgDn will move you a full screen at a time. Home and End will jump you to the extremes of the table.

Pressing N for Next or ENTER will advance you to the next view. If it does not exist you will be given the opportunity to create it.

Typing the number of an existing view, and then pressing ENTER, will return that view to the screen.

The Print command will print your table (provided your printer is properly connected) or write your table to a disk file. When you press

page 2 of 3

P you will see the following options:

title:
characters per line: 80
lines per page: 60
direct output to: PRN

You may type a title with up to 40 characters. The values for characters per line and lines per page may be modified by moving to the option fields with the arrow keys and typing in new values. The output is directed to the printer as a default; to write to a file on disk, type in the file name of your choice. ESC and Quit return you to View and DOS respectively.

Tabulate_All

This command will display all the variables that you have saved, in tabular format. The format of the table is the variable names listed on the left and the value of time shown at the top.

When you have invoked the Tabulate_All option, you will be queried for a print interval, the TIME interval between values. Either select the default by pressing Enter or type another value. The table will appear with a prompt line beneath that looks identical to the one described above for Select_Tabulate. Press L now for a description of that prompt line.

Esc

The ESC key (or invoking the command) will return you to the Viewer command level if pressed from a tabular or graphical display. When you press ESC at the View command level, you will be returned to PD top level.

QUIT:

The Quit command returns you directly to DOS regardless of where it was invoked.

```
:******************VARIABLE BATCH FILE*****************
bat /p /s * ... Loading ... SETTING MODEL VARIABLES
        Call -top1
        Exit
-top    goto -top%A

-top1 %A = 1
   BAT  color \1F
        ram
   BAT  cls
   BAT  begtype




                 \1A                  SET MODEL VARIABLES              \1F

   \1A                                                               \1F


                          \1D 1   \1F List Variables

                           \1D 2   \1F Definitions of Variables


                           \1D 3   \1F Set Variables
```

```
Choose an option:   (ESC exits menu):;
end
-1stkey1 inkey %0 | if %0 # = 1 type %0;
        if %0 = key01b return
        goto -%0~1
-2ndkey1 inkey %1 | if %1 # = 1 type %1;
        if %1 = key01b return
        if %1 = key020 goto -$%0$1
        if %1 = key00d goto -$%0$1
        if %1 = key008 goto -top1
        if %1 = key14b goto -top1
        goto -%0%11

-1~1 **** LISTING OF VARAIBLES ****
        bat Cls
        BAT COLOR \1F
        bat begtype
```

\1A LISTING OF VARIABLES \1F

The following 17 model variables of Dynamica are adjustable via the menu. They are grouped in 4 categories for easy access.

I. ACTUAL PROJECT SIZE

1. \1D RJBDSI \1F. . . . . Real Job Size in DSI

II. VARIABLES CHARACTERISTIC OF THE ORGANIZATIONAL ENVIRONMENT

A. Productivity

2. \1D DSIPTK \1F. . . . . . . Delivered Source Instruction Per Task

B. Quality

3. \1D TNERPK \1F. . Error Rate Per 1000 Delivered Source Instruction

C. Staffing Variables

4. \1D HIREDY \1F. . . . . Hiring Delay

5. \1D ASIMDY \1F. . . . . Assimilation Delay

6. \1D AVEMPT \1F. . . . . Average Employment

\1A Press <ENTER> To Continue \1F

end

inkey

bat cls

BAT COLOR \1F

bat begtype

\1A LISTING OF VARIABLES (CONT.) \1F

III. POLICY VARIABLES

A. Estimation

7. \1D UNDEST \1F. . . . . Task Underestimation Factor

8. \1D TOTMD1 \1F. . . . . Total Mandays

9. \1D TDEV1 \1F. . . . . Time to Develope

B. Resource Allocation

10. \1D DEVPRT \1F. . . . . % of Effort Assumed Needed For Development

11. \1D TPFMQA \1F. Fraction of Manpower Devoted to Quality Assurance

12. \1D INDUST \1F. . . . . Initial Understaffing Factor

13. \1D WCWF1 \1F. . . . . Willingness to Change the Workforce

C. Staffing

14. \1D TRPHNR \1F. . . % of Experienced Employee Effort to Train a New Employee

15. \1D AMPPS \1F. Average Daily Manpower / Staff Expended On Project

IV. MODEL CONTROL

16. \1D MAXLEN \1F. . . . . Max Length Of Project

17. \1D SAVPER \1F. . . . . Save Period For Data

```
    \1A          Press <ENTER> to return to the Menu      \1F
end
     inkey
     bat cls
   bat /p /s goto -top1

-2~1 **** DEFINITIONS OF VARIABLES ****
   call -top2
   bat /p /s goto -top1

-3~1 **** SETTING MODEL VARIABLES ****
     bat Cls
   dynex project.dnx
     bat Cls
   bat /p /s goto -top1
```

```
-%0~1
-$%0$1
-%0%11 beep goto -top1
-top2 %A = 2
        ram
     BAT  cls
     BAT COLOR \1F
     BAT  begtype
```

\1A DEFINITIONS OF VARIABLES \1F

\1D 1 \1F Actual Project Size \1F

\1D 2 \1F Organizational Environment Variables \1F

\1D 3 \1F Policy Variables \1F

\1D 4 \1F Model Control Variables \1F

```
Choose an option:    (ESC exits menu):;
end
-1stkey2 inkey %0 | if %0 # = 1 type %0;
          if %0 = key01b return
          goto -%0~2
-2ndkey2 inkey %1 | if %1 # = 1 type %1;
          if %1 = key01b return
          if %1 = key020 goto -$%0$2
          if %1 = key00d goto -$%0$2
          if %1 = key008 goto -top2
          if %1 = key14b goto -top2
          goto -%0%12

-1~2 **** ACTUAL PROJECT SIZE ****
       Bat cls
          BAT COLOR \1F
          bat begtype

                    \1A            ACTUAL PROJECT SIZE DEFINITION          \1F
\1D
REAL JOB SIZE IN DSI . . . . . . . . . . . . . . . . . . . . . . . . . . RJBDSI
\1F
```

The real size of the software project in delivered source instructions (DSI) with comments. The following definitions are from Boehm, 1981.

Delivered. This term is generally meant to exclude nondelivered support software such as test drivers. However, if these are developed with the same care as delivered software, with their own reviews, test plans, documentation, etc., then they should be counted.

Source Instructions. This term includes all program instructions created by project personnel and processed into machine code by some combination of preprocessors, compilers, and assemblers. It excludes comment cards and unmodified utility software. It includes job control language, format statements and data declarations. Instructions are defined as lines of code or card images. Thus, a line containing two or more source statements counts as one instruction; a five-line data declaration counts as five instructions.

```
    \1A               Press <ENTER> to return to the Menu          \1F

end
      inkey
      Bat *
    Bat cls
    bat /p /s goto -top2

-2~2 **** ORGANIZATIONAL ENVIRON VARIABL ****
      bat Cls
      BAT COLOR \1F
      bat begtype

        \1A         ORGANIZATIONAL ENVIRONMENT DEFINITIONS   \1F

\1D
DELIVERED SOURCE INSTRUCTION PER TASK . . . . . . . . . . . . . . . . . .
DSIPTK
\1F This parameter is set to the value of the nominal potential productivity
in the organizational environment being modeled. For example, if the nominal
potential productivity is 50 DSI/Man-Day then DSIPTK would be set to 50.
```

\1D

ERROR RATE PER 1000 DELIVERED SOURCE INSTRUCTIONS (KDSI) . . . . . . TNERPK

\1F TNERPK is the number of errors committed on the average, per 1000 DSI, as the software project is being developed. It is not a single number but rather a table function. This allows the user to set different error rates at different stages of the project's lifecycle.

\1D

HIRING DELAY . . . . HIREDY

\1F HIREDY is the average delay time, in work days, incurred in adding new staff members to the project.

\1A Press <ENTER> to Continue \1F

end

inkey

bat cls

BAT COLOR \1F

bat begtype

\1A ORGANIZATIONAL ENVIRONMENT DEFINITIONS (CONT.) \1F

\1D

HIRING DELAY . . . . HIREDY

\1F HIREDY is the average delay time, in work days, incurred in adding new staff members to the project.

\1D

ASSIMILATION DELAY . . . . ASIMDY

\1F ASIMDY is the average time needed to assimilate new hirees into the project, measured in workdays. It is the time required for orientation, institutionalization and training.

\1D

AVERAGE EMPLOYMENT . . . . AVEMPT

\1F AVEMPT is the average employment time of project team members, measured in working days. It is indirectly proportional with TURNOVER. For example, the smaller the AVEMPT the larger the turnover rate, and conversely, the smaller the turnover rate the higher the AVEMPT.

\1A Press <ENTER> to return to the Menu \1F

```
end
        inkey
        bat cls
        bat /p /s goto -top2

-3~2 **** POLICY VARIABLES ****
        bat Cls
        bat begtype
```

\1A POLICY DEFINITIONS \1F

\1D

TASK UNDER-ESTIMATION FACTOR . . . . UNDEST

\1F Undersizing is a large problem in project management. This variable allows the user to experiment with different values of underestimates. Thus to simulate a situation where the project size is initially underestimated by 25 %, this variable would be set to 0.25.

\1D

TOTAL MAN DAYS TOTMD1

\1F TOTMD1 is the man-day estimate produced at the start of the project, for design, coding and system testing of the project.

\1D

TIME TO DEVELOP TDEV1

\1F TDEV1 represents the estimated time, measured in work days, for the design, coding and testing stages of the project, produced at the start of the project. For example, if the initial project duration estimate was 16 months, TDEV1 would be equal to (16 x 20) or 320 working days.

\1A Press <ENTER> to Continue \1F

end

inkey

bat cls

bat begtype

\1A POLICY DEFINITIONS (CONT.) \1F

\1D

% OF EFFORT ASSUMED NEEDED FOR DEVELOPMENT . . . . DEVPRT

\1F This variable is used to allocate the projected total budget, measured in Man-Days, for the development (which includes design and coding) and testing phases. For example, if it is decided that 80 % of the total MAn-Days would be allocated tc development and 20 % to testing, then DEVPRT would be set to 0.80.

\1D

FRACTION OF MANPOWER DEVOTED TO QUALITY ASSURANCE . . . . TPFMQA

\1F TPFMQA is the percent of development effort allocated in the project's plan for QA activities during the design and coding stages. It is not a single number but rather a table function that allows the user to change the number of Man-Days allocated to QA at different stages in the project lifecycle. A value of 0.20 at some point in the project would indicate that at this point, 20 % of the Man-Day budget is allocated to QA.

\1A Press <ENTER> to Continue \1F

end

inkey

bat cls

bat begtype

\1A POLICY DEFINITIONS (CONT.) \1F

\1D

INITIAL UNDERSTAFFING FACTOR . . . . INDUST

\1F The project's average staffing level is equal to the project's total number od Man-Days divided by the project's scheduled duration. Typically, the project will start with a small core team which grows in size. This

variable specifies the size of the starting group. Thus a value of 0.5 indicates that the startup team is half the size of the average staff size.

\1A Press <ENTER> to continue \1F

end

inkey

bat cls

bat begtype

\1A POLICY DEFINITIONS (CONT.) \1F

\1D

WILLINGNESS TO CHANGE THE WORKFORCE . . . .

.WCWF1

\1F On deciding upon a "Workforce" level desired, project managers typically consider a number of factors. One important factor is the project's scheduled completion date. As part of the planning function, management determines the workforce level that it believes is necessary to complete the project on schedule. In addition to this factor, consideration is also given to the stability of the workforce. Thus, before adding new project members, management tries to contemplate the project employment for new members. Different organizations weigh this factor differently. In general, the relative weighing between th desire for workforce stability on one hand and

the desire to complete the project on time, on the other, is not static, but changes dynamically throughout the lifecycle of the project. For example, toward the end of the project there is typically considerable reluctance to bring in new people, even if the project is behind schedule. It would take too much time and effort (relative to the time and effort that are remaining) to acquaint new people with the mechanics of the project, integrate them into the project team and train them in the necessary technical areas.

\1A Press <ENTER> for More Information of WCWF1 \1F

END

INKEY

bat cls

bat begtype

\1A POLICY DEFINITIONS (CONT.)

WCWF1 DEFINITION (CONT.) \1F

These managerial considerations are operational in the model as follows:

WORKFORCE LEVEL NEEDED = (INDICATED WORKFORCE LEVEL) * (WCWF) +

(CURRENT WORKFORCE) * (1-WCWF)

The Weighting Factor (WCWF) is termed Willingness to Change the Workforce.

In the early stages of the project when "Time Remaining" is generally much larger than the sum of "Hiring Delay" and the "Average Assimilation Delay" WCWF would be equal to 1. When WCWF is = 1, the "Workforce Level Needed" in

the above equation would simply be equal to the "Indicated Workforce Level," i.e., management would be adjusting its workforce size to the level it feels is needed to finish on schedule. The "Indicated Workforce Level" can be

determined by dividing the amount of effort that management percieves is still remaining (in Man-Days) by the time remaining to complete the project in days.

When the "Time Remaining" decreases, ASSIMILATION DELAY (ASIMDY) could

be set to equal exactly 0. The "Workforce Level Needed" in the equation would thus be equal to the "Current Workforce" i.e., management attempts to maintain the projects workforce at its current level, and make adjustments to the schedule instead.

\1A Press <ENTER> to Continue \1F

end

inkey

bat cls

bat begtype

\1A POLICY DEFINITIONS (CONT.) \1F

\1D

TRAINERS PER NEW EMPLOYEE . . . . TRPHNR

\1F In most organizations, training of new employees is carried out by the more experienced employees. This variable defines the fraction of an experienced staff member's time that is devoted to train new hirees. For example, a value of 0.2 indicates that on the average each new employee consumes 20 % of an experienced employee's time for the duration of the assimilation delay.

\1D

AVERAGE DAILY MANPOWER PER STAFF EXPENDED ON PROJECT . . . . ADMPPS

\1F Project members are often only assigned part-time to a project. ADMPPS defines the % of time that the team members devote, on an average to the

particular project being simulated. For example, if staff members tend to divide their time equally between two projects, i.e., half of their time is spent on project A and the other half is spent on project B, then ADMPPS would be = 0.5.

\1A Press <ENTER> to Continue \1F

```
end
    inkey
    bat Cls
  bat /p /s goto -top2

-4~2 **** MODEL CONTROL VARIABLES ****
    BAT Cls
    BAT BEGTYPE
```

\1A MODEL CONTROL DEFINITIONS \1F

\1D

MAX LENGTH OF PROJECT . . . . MAXLEN

\1F This model control variable is useful in controlling the simulations. It precludes the model from continuing a simulation beyond a predetermined length, e.g., 1000 days.

\1D

SAVE PERIOD FOR DATA . . . . SAVPER

\1F This model control varaible allows the user to adjust the models save period for data for later viewing. For example, if the user sets the SAVPER = 1, the model will save all variable values on a daily basis, this data would be available for the user to examine and plot at the end of a simulation. A SAVPER of 10, saves values every 10 days. This provides the

user with a trade off between the time required for a simulation and the level of detail of the results.

```
                    \1A        Press <ENTER> to return to the Menu        \1F
END
          bat inkey
          bat Cls
       bat /p /s goto -top2

-%0~2
-$%0$2
-%0%12 beep goto -top2
-on.error-
 if %R > 82 if %R < 90 type !! Floating Point Error !! |goto -Calc.
 Cls beep type Unexpected batch file error %R in line %L |exit
```

```
:********************PLOT BATCH FILE*********************
BAT /P /S
    CALL -top1
    exit
-top1
       color \1F
       ram
       cls
       begtype
```

\1A PLOTTING RESULTS \1F

\1A THE FOLLOWING MENU ALLOWS THE USER TO VIEW AND SAVE 4 PREDEFINED PLOTS. \1F

\1D 0 \1F OVERVIEW PLOTTING FUNCTIONS

\1D 1 \1F PLOT 1

\1D 2 \1F PLOT 2

```
                 \1D 3   \1F PLOT 3


                 \1D 4   \1F PLOT 4



Choose an option:   (ESC exits menu):;
end
-1stkey1 inkey %0 | if %0 # = 0 type %0;
        if %0 = key01b return
        bat goto -%0~1
-2ndkey1 inkey %1 | if %1 # = 0 type %1;
        if %1 = key01b return
        if %1 = key020 goto -$%0$1
        if %1 = key00d goto -$%0$1
        if %1 = key008 goto -top1
        if %1 = key14b goto -top1
        goto -%0%11

-0~1 **** OVERVIEW PLOTTING FUNCTIONS ****
      BAT CLS
        BAT COLOR \1F
BAT BEGTYPE



*****************************************************************************
*******
                OVERVIEW OF PLOTTING FUNCTION
```

```
*****************************************************************************
*******


        THE DYNAMICA MODEL CAN CREATE PLOTS OF PRESELECTED
    VARIABLES.  EACH PLOTTING FUNCTION, CONTAINS A LIST OF THE
    VARIABLES PLOTTED BY THAT MENU SELECTION.  VIEWING A
PLOT IS
    SIMPLY A MATTER OF SELECTING THE NUMBER OF THE PLOT
DESIRED.
    THESE PLOTS MAY BE PRINTED USING THE PRINT FUNCTION OF
THE
    INCLUDED MENU.

        IF THE USER NEEDS A PRINTOUT OF A GRAPH HE CAN PRINT
THE SCREEN
    WITH THE PRTSC FUNCTION.

                HIT ANY KEY TO CONTINUE !

END

BAT INKEY
BAT CLS

      bat /p /s goto -top1

-1~1 **** PLOT 1 ****
     BAT CLS
BAT COLOR \1F
BAT BEGTYPE
```

```
*******************************************************************************
*******
  \1A                         PLOT 1       \1F
*******************************************************************************
*******

     PLOT 1 GRAPHS THE FOLLOWING VARIABLES:

          SCHCDT . . . . . . . . . . . ESTIMATED SCHEDULE IN DAYS
          PJBSZ . . . . . . . . . . PERCEIVED PROJECT SIZE IN TASKS
          JBSZMD . . . . . . . ESTIMATED PROJECT COST IN MAN-DAYS
          TOTWF . . . . . . . . . . . . . . TOTAL WORKFORCE PEOPLE
          CUMMD . . . . . . . . . . . CUMULATIVE MAN-DAYS EXPENDED


  \1A       IF YOU HAVE AN EGA CARD TYPE (Y)ES IF NOT TYPE (N)O.
\1F


END

BAT INKEY  %0

BAT CLS
BAT IF Y = %0 THEN GOTO -EGA1
BAT ELSE GOTO -OTHER1
```

```
BAT -EGA1
REP PROJECT PLOT1
BAT GOTO -NEXT1
BAT -OTHER1
REP PROJECT PLOT1 -PLM 6
BAT -NEXT1
bat /p /s goto -top1

-2~1 **** PLOT 2 ****
     BAT CLS
BAT COLOR \1F
BAT BEGTYPE
```

```
*****************************************************************************
*******
  \1A                    PLOT 2          \1F
*****************************************************************************
*******
        FIRST, A QUICK REVIEW OF THE VARIABLES BEING PLOTTED:

CMTKDV . . . . . . . . . . . . . . . . . . . CUMULATIVE TASKS DEVELOPED
CUMTKT . . . . . . . . . . . . . . . . . . . .CUMULATIVE TASKS TESTED
CUMMD  . . . . . . . . . . . . . . . . . . CUMULATIVE MAN-DAYS EXPENDED
PJBSZ  . . . . . . . . . . . . . . . PERCEIVED PROJECT SIZE IN TASKS
PDEVRC . . . . . . . . . . . . . . . ESTIMATED % DEVELOPMENT COMPLETE
```

```
 \1A        IF YOU HAVE AN EGA CARD TYPE (Y)ES IF NOT TYPE (N)O.
\1F



END
BAT INKEY %0

BAT CLS
BAT IF Y = %0 THEN GOTO -EGA2
BAT ELSE GOTO -OTHER2

BAT -EGA2
REP PROJECT PLOT2
BAT GOTO -NEXT2
BAT -OTHER2
REP PROJECT PLOT2 -PLM 6
BAT -NEXT2
bat /p /s goto -top1

-3~1 **** PLOT 3 ****
     BAT CLS
BAT color \1F
BAT BEGTYPE




*******************************************************************************
*******
                                  PLOT 3
```

```
*****************************************************************************
*******

          FIRST, A QUICK REVIEW OF THE VARIABLES BEING PLOTTED:

TOTWF . . . . . . . . . . . . . . . . . . . . . . . . TOTAL WORKFORCE
FRWFEX . . . . . . . . .  FRACTION OF WORKFORCE THAT IS EXPERIENCED
SDVPRD . . . . . . . . . . . . . . . . . . . . . . . . . PRODUCTIVITY
COMMOH . . . . . . . . . . . . . . . . . . . . . COMMUNICATION OVERHEAD


          IF YOU HAVE AN EGA CARD TYPE (Y)ES IF NOT TYPE (N)O.


END
BAT INKEY %0

BAT CLS
BAT IF Y = %0 THEN GOTO -EGA3
BAT ELSE GOTO -OTHER3

BAT -EGA3
REP PROJECT PLOT3
BAT GOTO -NEXT3

BAT -OTHER3
REP PROJECT PLOT3 -PLM 6
```

```
BAT -NEXT3
bat /p /s goto -top1

-4~1 **** PLOT 4 ****
     BAT CLS
BAT COLOR \1F
BAT BEGTYPE
```

```
******************************************************************************
*******
  \1A                    PLOT 4       \1F
******************************************************************************
*******
        FIRST, A QUICK REVIEW OF THE VARIABLES BEING PLOTTED:

AFMPDJ . . . . . . . . . .  ACTUAL FRACTION OF A MAN-DAY ON PROJECT
JBSZMD . . . . . . . . . . . . PERCEIVED TOTAL JOB SIZE IN MAN-DAYS
PJBSZ  . . . . . . . . . . . . . . . . . .PERCEIVED JOB SIZE IN TASKS
PMDSHR  . . . . . . . . . . . . . .  PERCEIVED SHORTAGE IN MAN-DAYS



 \1A     IF YOU HAVE AN EGA CARD TYPE (Y)ES IF NOT TYPE (N)O.
\1F



END
```

```
BAT INKEY %0

BAT CLS
BAT IF Y = %0 THEN GOTO -EGA4
BAT ELSE GOTO -OTHER4

BAT -EGA4
REP PROJECT PLOT4
BAT GOTO -NEXT4

BAT -OTHER4
REP PROJECT PLOT4 -PLM 6
BAT -NEXT4
bat /p /s goto -top1

-%0~1
-$%0$1
-%0%11 beep goto -top1
```

```
:**************STORE STATISTICS BATCH FILE***************
BAT /P /S
        BAT COLOR \1F
        RAM
        BAT CLS
        ERASE PROJECT.OUT
        BAT COLOR \1F
        BAT CLS
        REP PROJECT STATS

         BAT COLOR \1F
         BAT CLS
         %0 = PROJECT.OUT
         <%0
         -readloop
         read.parsed %A
         if %A = ^Z skip 1
         Type %A | goto -readloop
         <
BAT BEGTYPE

         \1A                PRESS \1D<RETURN> \1Ato Continue\1F
end
        inkey
        BAT
        BAT CLS
        BAT COLOR \1F
BAT BEGTYPE
```

PLEASE ENTER THE FILE NAME YOU DESIRE FOR THIS
REPORT.

NOTE:

CHANGES MADE TO VARIABLES ARE STORED IN THE FILE.
FOR

EXAMPLE IF HIRING DELAY WAS CHANGED FROM 30 TO 40,
THAT

INFORMATION WOULD BE STORED AT THE END OF THE
REPORT.

```
END
BAT LOCATE 14 11
BAT READ %0
BAT CLS
BAT COLOR \1F
BAT BEGTYPE
```

ENTER THE FULL PATH NAME OF THE DIRECTORY WHERE YOU WOULD LIKE THE RESULTS STORED.

or

PRESS <RETURN> TO ACCEPT THE DEFAULT DIRECTORY

```
END
    BAT LOCATE 11 18
    BAT READ %1
    BAT CLS
    ERASE %0
    BAT COLOR \1F
    BAT CLS
    COPY PROJECT.OUT %0
    BAT COLOR \1F
    BAT CLS
    COPY %0 %1
    BAT COLOR \1F
    BAT CLS
    EXIT
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Director, Information Systems (OP-945) Office of the Chief of Naval Operations Navy Department Washington, D. C. 20350-2000	1
4.	Commandant of the Marine Corps Code TE 06 Headquarters, U.S. Marine Corps Washington, D. C. 20360-0001	2
5.	Superintendent, Naval Postgraduate School Computer Technology Programs, Code 37 Monterey, California 93943-5000	1
6.	Professor Tarek Abdel-Hamid, Code 54AH Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	5
7.	Professor Magdi Kamel, Code 54KA Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
8.	Maj Daniel W. Swindell 9046 Baber Drive Manassas, Virginia 22110	2